

AD 696.567

NPS-58DW9071A

UNITED STATES NAVAL POSTGRADUATE SCHOOL



NOV 1 1969

DIGITAL ANALYSIS OF TURBULENCE DATA
ON THE IBM 360/67 AT THE
NAVAL POSTGRADUATE SCHOOL

by

J. R. Wilson
N. E. J. Boston
W. W. Denner

1 July 1969

This document has been approved for public
release and sale; its distribution is unlimited.

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va 22151

100

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Robert W. McNitt, USN
Superintendent

R. F. Rinehart
Academic Dean

ABSTRACT:

A system of time series programs used by the Institute of Oceanography of the University of British Columbia was made available to the Department of Oceanography of the Naval Postgraduate School in February 1969. This report summarizes the system and outlines the procedures to be followed in using the programs.

The system consists of three programs labelled UBC FTOR, UBC SCOR and UBC FC PLOT. The program UBC FTOR computes Fourier coefficients from selected channels of analog-to-digital tape and writes them on another tape. The program UBC SCOR reads the tape produced by UBC FTOR and from the Fourier coefficients calculates spectra, cospectra and quadrature spectra for the channels indicated. These are computed for each data block. The printed output gives for each quantity the average, standard deviation and a number representing the trend over the blocks. In the case of co- and quad-spectra phase and coherence are also printed out. The program UBC FC PLOT provides a Calcomp plot of the spectra for qualitative analysis.

These programs have been tested on the IBM 360/67 of the Naval Postgraduate School and produced for a test tape the same answers as produced by the U.B.C. machine.

A system to develop the capability to use the SDS-9300 and the associated analog computer available at the Naval Postgraduate School to digitize data to be analyzed by the time series programs is included as Appendix I.

This task was supported by: Naval Ordnance Systems Command
Navy Department

Noël E. J. Boston
N. E. J. Boston
Assistant Professor of Oceanography

Approved by:

ORIGINAL SIGNED BY
D. F. Leipper
Chairman
Department of Oceanography

W. W. Denner
W. W. Denner
Assistant Professor of Oceanography

Released by:

C. E. Menneken
C. E. Menneken
Dean of Research Administration

NPS-58DW9071A

NAVAL POSTGRADUATE SCHOOL

Department of Oceanography
Monterey, California

DIGITAL ANALYSIS OF TURBULENCE DATA ON THE IBM 360/67
AT THE NAVAL POSTGRADUATE SCHOOL

by

J. R. Wilson

N. E. J. Boston

W. W. Denner

This research was sponsored by Naval Ordnance Systems Command under NAVORD CONTRACT ORD TASK-03C-005/551-1/UR104-03-01. The work reported herein is of a preliminary nature and the results are not necessarily in final form. Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

July 1969

FORWARD

The digital programs discussed in this report were developed at the University of British Columbia by John Garrett and Ron Wilson during 1967. They have been applied to a large variety of geophysical turbulence data and found extremely successful. Because of the nature of the air/sea interaction program being carried out by the Department of Oceanography of the Naval Postgraduate School, such programs were desirable to be available at the computing facilities of the School.

In February of 1969, Mr. Ron Wilson was hired as a consultant to convert the University of British Columbia programs for use on the IBM 360/67 computer of the Naval Postgraduate School. This report is a summary of the system based on notes provided by Mr. Wilson. The principal investigators take full responsibility for any errors or omissions that may occur in this report.

Noel Boston
W. W. Denner
Principal Investigators

ABSTRACT

A system of time series programs used by the Institute of Oceanography of the University of British Columbia was made available to the Department of Oceanography of the Naval Postgraduate School in February 1969. This report summarizes the system and outlines the procedures to be followed in using the programs.

The system consists of three programs labelled UBC FTOR, UBC SCOR and UBC FCPILOT. The program UBC FTOR computes Fourier coefficients from selected channels of analog-to-digital tape and writes them on another tape. The program UBC SCOR reads the tape produced by UBC FTOR and from the Fourier coefficients calculates spectra, cospectra and quadrature spectra for the channels indicated. These are computed for each data block. The printed output gives for each quantity the average, standard deviation and a number representing the trend over the blocks. In the case of co- and quad-spectra phase and coherence are also printed out. The program UBC FCPILOT provides a Calcomp plot of the spectra for qualitative analysis.

These programs have been tested on the IBM 360/67 of the Naval Postgraduate School and produced for a test tape the same answers as produced by the U. B. C. machine.

A system to develop the capability to use the SDS-9300 and the associated analog computer available at the Naval Postgraduate School to digitize data to be analyzed by the time series programs is included as Appendix I.

CONTENTS

	Page
FORWARD	3
ABSTRACT	4
LIST OF FIGURES	6
LIST OF TABLES	7
I. GENERAL	8
II. DIGITAL SAMPLING PROCEDURE	9
A. Analog Signal Level	9
B. Aliasing	9
C. Sampling Rate	10
D. Miscellaneous Factors	11
III. COMPUTATION OF SPECTRA AND CROSS SPECTRA	12
IV. INPUT DATA TAPE FORMAT	25
V. PROCEDURE IN ANALYZING DATA	27
VI. SOURCE PROGRAMS	28
ACKNOWLEDGEMENTS	29
REFERENCES	30
APPENDIX I ANALYZING DATA DIGITIZED ON THE SDS-9300	31
APPENDIX II FORTRAN SOURCE PROGRAMS	33

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1	Digital Sampling Procedure	10
2	FCPLOT of DISA U Spectrum	20
3	FCPLOT of Sonic W Spectrum	21
4	FCPLOT of UW Co-spectrum	22
5	FCPLOT of UW Quadrature Spectrum	23
6	FCPLOT of Fourier Coefficients for DISA U Data	24

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
I	DISA U Spectrum	15
II	Sonic W Spectrum	16
III	UW Co-spectrum	17
IV	UW Quadrature Spectrum	18
V	Fourier Coefficients for DISA U data	19

I. GENERAL

This report is intended (a) to summarize the system of time series programs used by the Institute of Oceanography of the University of British Columbia which are now available to the Department of Oceanography of the Naval Postgraduate School and (b) to describe briefly the procedures to be followed in using them.

These programs were made available in February 1969 and this report describes the programs as they existed at that time. Some updating and expansion of these programs may occur from time to time. A check on their current state is recommended prior to using.

The turbulence data used to test these programs included downstream velocity fluctuations measured with a hot-wire anemometer (DISA U) and vertical velocity fluctuations measured with a sonic anemometer (Sonic W). These measurements were made simultaneously during August 1968 at the U.B.C. Institute of Oceanography experimental site at Spanish Banks, British Columbia.

II DIGITAL SAMPLING PROCEDURE

The system is based on the assumption that the data to be analyzed are initially recorded on analog tape. These analog data must then be converted to digital form before they can be processed by a digital computer. The analog to digital procedure is of central importance and must be done with extreme care. Almost always there must be some conditioning of the signal prior to the analog-to-digital (A/D) conversion. Common problems to consider are analog signal level, aliasing and sampling frequency. Of course each analysis problem presents its unique difficulties and must be treated individually.

A. Analog Signal Level

The patch of signal to be analyzed may not have been recorded under ideal conditions. It may vary from almost too small (approaching system noise level) to almost too large (occasional clipping). To insure good digitizing some amplification may be necessary to provide the optimum signal input to the A/D converter. If there is an appreciable DC level on the signal, high pass filtering or rerecording may be necessary before digitization can begin. In brief, the signal level must always be checked to be sure maximum advantage is being taken of the full dynamic range of the A/D converter.

B. Aliasing

When an analog signal is sampled at a frequency F , energy present in the signal at frequencies $f > \frac{F}{2}$ will appear in the sampled output at an apparent (aliased) frequency $F-f$. This phenomenon is known as aliasing

with $\frac{F}{2}$ called the folding or Nyquist frequency (Blackman & Tukey, 1951).

To prevent aliasing, the input to the analog-to-digital converter must contain no energy at frequencies greater than the folding frequency associated with the particular sampling rate being used. This can be done by using an analog low pass filter between the signal source and the converter input. The filter cutoff frequency is normally less than the folding frequency. In practice there is no exact relation between these frequencies as it depends on the input levels expected at high frequencies and the sharpness of the filter cutoff.

C. Sampling Rate

If three channels are to be sampled and co-spectra are required between all combinations the sampling procedure might be as illustrated in Figure 1.

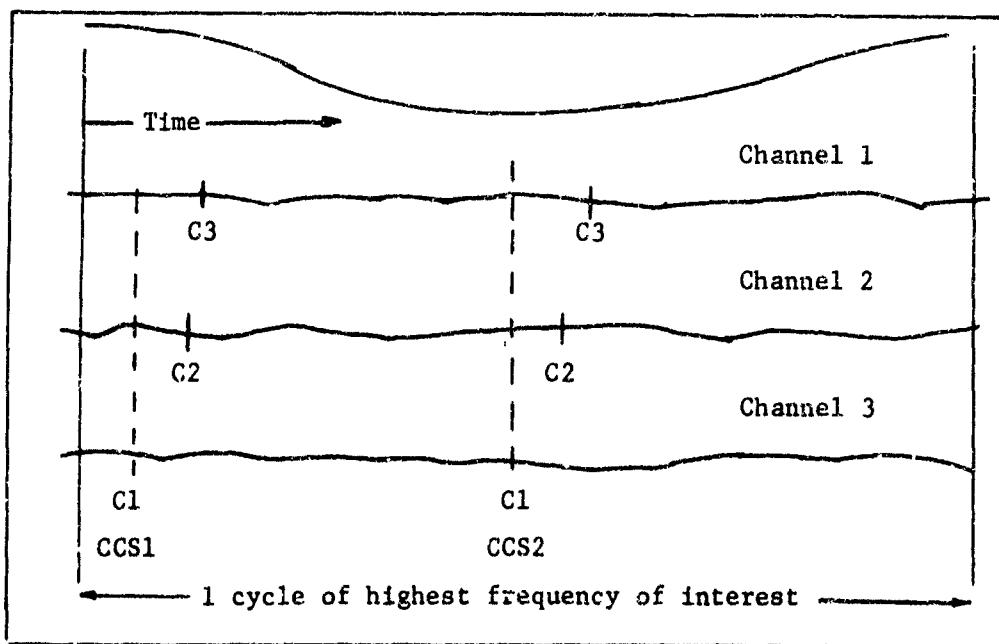


Figure 1. Sampling Procedure

The cuts C1, C2 and C3 represent the digital samplings of channels 1, 2 and 3 respectively.

CCS1 and CCS2 represent two samplings of all channels and are referred to as Cross Channel Sweeps.

The time lag between sampling channel 1 and channel 3 in a cross channel sweep should be a small fraction of a cycle since the program assumes they are sampled simultaneously for purposes of computing the co-spectra and quadrature spectra.

The time between cross channel sweeps should be such that each channel is sampled at least twice per cycle of the highest frequency with sufficient amplitude to produce an aliasing problem.

D. Miscellaneous Factors

It is important that all factors influencing the validity and interpretations of results be considered prior to digitizing. Besides those factors already discussed these should include the record length, the block length, spectral windows and how these will effect the stability of the spectral estimates. Further, an estimate of the computer time required for the analysis should be made. There has not been sufficient experience with the computing facilities at the Naval Postgraduate School to allow tables of time for typical calculations to be presented. However the tables given by Garrett (1967) will indicate order of magnitude computing times. In addition his report contains excellent discussions of some of the factors mentioned here.

III COMPUTATION OF SPECTRA AND CROSS SPECTRA

The computation of the spectral densities employ the Fast Fourier transform (FFT) of Cooley and Tukey (1965). The program used is distributed through the share library and is catalogued at PKFORT SDA3465.

The subroutine will transform a number of data points equal to 2^n where n is a positive integer less than or equal to 13. This means the largest number of data points that can be transformed is 8192. Since the time series encountered in turbulence measurements generate more points than this, some method of transforming and averaging consecutive blocks of data must be used. This also makes it possible to gain some information as to the variance of the spectral or cospectral densities over the interval of the signal.

The method employed consists of evaluating the spectrum, co-spectrum or quadrature spectrum for each frequency band for successive blocks of 2^n data points. The mean and variance of these densities are then formed and printed and plotted.

For example purposes, assume 100 blocks of 2048 points each are transformed. For each block there will be contained in the Fourier coefficients an amplitude and phase for frequencies corresponding to 0, 1, 2, ..., 1024 cycles over the 2048 points. Zero is the mean and 1024 is the folding frequency. Therefore for each block of 2048 data points there are 1024 estimates of spectral, cospectral or quadrature spectral density. This is considered to be too much information to plot graphically so in the production of printouts and plots further averaging than along block must be done. For example the program may divide the spectrum into 32 bands and average across these bands as well as along blocks to obtain the mean and variance of the quantities.

If there are two channels and they have for the i^{th} harmonic, amplitudes A_i and B_i , and phases α_i and β_i , then the spectral densities are

$$\frac{A_i^2}{2 \times \text{Bandwidth}} \quad \text{and} \quad \frac{B_i^2}{2 \times \text{Bandwidth}}$$

where the bandwidth is given by

$$\frac{\text{sampling frequency}}{\text{no. of data points in the transform}}$$

The cospectrum is given by

$$\frac{A_i \times B_i \cos(\alpha_i - \beta_i)}{2 \times \text{Bandwidth}}$$

The quadrature spectrum is given by

$$\frac{A_i \times B_i \sin(\alpha_i - \beta_i)}{2 \times \text{Bandwidth}}$$

These quantities are computed for each frequency and block and are then averaged across the frequencies and along blocks.

The error bars plotted are computed by evaluating the variance of the estimates of the spectral density in the band and dividing by the square root of the number of estimates to establish the expected variance in the mean. In the case of the SCOR program the plotted and printed error bars represent standard deviations in the mean. The FCPLT program uses 1.96 times the standard deviation (or some other kind of 95% confidence interval).

The values of phase and coherence produced by the SCOR program are derived directly from the final values of the co- and quadrature spectra appearing to their left (Table III, for example).

Further information is available on the comments cards with the Fortran source programs (Appendix II).

Examples of print out for spectra, co-spectrum, quadrature spectrum and Fourier coefficients are given in Tables I through V. The corresponding FCPILOTS for these tables are Figures 2 through 6.

TABLE I

SPECIFY STATION FOR 1C2 BLOCKS CHANNEL 2 SAMPLES YARD SONIC TAPE 147/1/10/9/68
 SYSTEM SAMPLING FREQUENCY WAS 62.771 SAY P/S SEC MAKING THE BLOCK LENGTH 16.31726 SECONDS
 TREND IS THE AVERAGE OF 1 VALUE (A1) - VALUE (A1) / BLOCK NO. (A1-BLOCK NO.)
 A CALIBRATION FACTOR OF -P.C50E-C1 HAS BEEN APPLIED TO THE INPUT DATA

LAST HARMONIC			
1.122346804561762		1.122346804561762	
1.122346804561762		1.122346804561762	
1.122346804561762		1.122346804561762	
SPECTRUM	LAST HARMONIC	STD. DEV.	TREND
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
SPECTRUM	LAST HARMONIC	STD. DEV.	TREND
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
SPECTRUM	LAST HARMONIC	STD. DEV.	TREND
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
FREQUENCY BANDWIDTH	LAST HARMONIC	STD. DEV.	TREND
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
FREQUENCY HERTZ	LAST HARMONIC	STD. DEV.	TREND
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
INTEGRAL (SUM)	LAST HARMONIC	STD. DEV.	AVERAGE
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762
1.122346804561762	1.122346804561762	1.122346804561762	1.122346804561762

TABLE II

SPECIFICATIONS STATED FOR 162 BLOCKS CHANNEL 24 SAMPLED EACH TAPE 140/1/10/9/68
 THE SAMPLING FREQUENCY WAS 62.77105A P/SEC TAKING THE BLOCK LENGTH 16.31326 SECONDS
 TREND IS THE AVERAGE OF (A) - VALUE (A) - CLOCK AND (B) - CLOCK NO. (181)
 A CALIBRATION FACTOR OF 6.54CE-01 HAS BEEN APPLIED TO THE INPUT DATA

FREQUENCY HERTZ	BANDWIDTH	SPECTRUM MH/SEC	STD. DEV. 2/HERTZ	TREND	FREQ*SPECTRUM MH/SEC	LAST HARMON
5.0	3.1E-02	1.02	1.7E-01	-0.84E-04	-0.84E-04	
11.0	1.2E-02	1.02	0.0E-02	-1.43E-05	-1.43E-05	
17.0	1.3E-02	1.02	0.0E-02	-2.5E-05	-2.5E-05	
23.0	1.4E-02	1.02	0.0E-02	-3.6E-05	-3.6E-05	
29.0	1.5E-02	1.02	0.0E-02	-4.7E-05	-4.7E-05	
35.0	1.6E-02	1.02	0.0E-02	-5.8E-05	-5.8E-05	
41.0	1.7E-02	1.02	0.0E-02	-6.9E-05	-6.9E-05	
47.0	1.8E-02	1.02	0.0E-02	-8.0E-05	-8.0E-05	
53.0	1.9E-02	1.02	0.0E-02	-9.1E-05	-9.1E-05	
59.0	2.0E-02	1.02	0.0E-02	-1.02E-04	-1.02E-04	
65.0	2.1E-02	1.02	0.0E-02	-1.13E-04	-1.13E-04	
71.0	2.2E-02	1.02	0.0E-02	-1.24E-04	-1.24E-04	
77.0	2.3E-02	1.02	0.0E-02	-1.35E-04	-1.35E-04	
83.0	2.4E-02	1.02	0.0E-02	-1.46E-04	-1.46E-04	
89.0	2.5E-02	1.02	0.0E-02	-1.57E-04	-1.57E-04	
95.0	2.6E-02	1.02	0.0E-02	-1.68E-04	-1.68E-04	
101.0	2.7E-02	1.02	0.0E-02	-1.79E-04	-1.79E-04	
107.0	2.8E-02	1.02	0.0E-02	-1.90E-04	-1.90E-04	
113.0	2.9E-02	1.02	0.0E-02	-2.01E-04	-2.01E-04	
119.0	3.0E-02	1.02	0.0E-02	-2.12E-04	-2.12E-04	
125.0	3.1E-02	1.02	0.0E-02	-2.23E-04	-2.23E-04	
131.0	3.2E-02	1.02	0.0E-02	-2.34E-04	-2.34E-04	
137.0	3.3E-02	1.02	0.0E-02	-2.45E-04	-2.45E-04	
143.0	3.4E-02	1.02	0.0E-02	-2.56E-04	-2.56E-04	
149.0	3.5E-02	1.02	0.0E-02	-2.67E-04	-2.67E-04	
155.0	3.6E-02	1.02	0.0E-02	-2.78E-04	-2.78E-04	
161.0	3.7E-02	1.02	0.0E-02	-2.89E-04	-2.89E-04	
167.0	3.8E-02	1.02	0.0E-02	-3.00E-04	-3.00E-04	
173.0	3.9E-02	1.02	0.0E-02	-3.11E-04	-3.11E-04	
179.0	4.0E-02	1.02	0.0E-02	-3.22E-04	-3.22E-04	
185.0	4.1E-02	1.02	0.0E-02	-3.33E-04	-3.33E-04	
191.0	4.2E-02	1.02	0.0E-02	-3.44E-04	-3.44E-04	
197.0	4.3E-02	1.02	0.0E-02	-3.55E-04	-3.55E-04	
203.0	4.4E-02	1.02	0.0E-02	-3.66E-04	-3.66E-04	
209.0	4.5E-02	1.02	0.0E-02	-3.77E-04	-3.77E-04	
215.0	4.6E-02	1.02	0.0E-02	-3.88E-04	-3.88E-04	
221.0	4.7E-02	1.02	0.0E-02	-3.99E-04	-3.99E-04	
227.0	4.8E-02	1.02	0.0E-02	-4.10E-04	-4.10E-04	
233.0	4.9E-02	1.02	0.0E-02	-4.21E-04	-4.21E-04	
239.0	5.0E-02	1.02	0.0E-02	-4.32E-04	-4.32E-04	
245.0	5.1E-02	1.02	0.0E-02	-4.43E-04	-4.43E-04	
251.0	5.2E-02	1.02	0.0E-02	-4.54E-04	-4.54E-04	
257.0	5.3E-02	1.02	0.0E-02	-4.65E-04	-4.65E-04	
263.0	5.4E-02	1.02	0.0E-02	-4.76E-04	-4.76E-04	
269.0	5.5E-02	1.02	0.0E-02	-4.87E-04	-4.87E-04	
275.0	5.6E-02	1.02	0.0E-02	-4.98E-04	-4.98E-04	
281.0	5.7E-02	1.02	0.0E-02	-5.09E-04	-5.09E-04	
287.0	5.8E-02	1.02	0.0E-02	-5.20E-04	-5.20E-04	
293.0	5.9E-02	1.02	0.0E-02	-5.31E-04	-5.31E-04	
299.0	6.0E-02	1.02	0.0E-02	-5.42E-04	-5.42E-04	
305.0	6.1E-02	1.02	0.0E-02	-5.53E-04	-5.53E-04	
311.0	6.2E-02	1.02	0.0E-02	-5.64E-04	-5.64E-04	
317.0	6.3E-02	1.02	0.0E-02	-5.75E-04	-5.75E-04	
323.0	6.4E-02	1.02	0.0E-02	-5.86E-04	-5.86E-04	
329.0	6.5E-02	1.02	0.0E-02	-5.97E-04	-5.97E-04	
335.0	6.6E-02	1.02	0.0E-02	-6.08E-04	-6.08E-04	
341.0	6.7E-02	1.02	0.0E-02	-6.19E-04	-6.19E-04	
347.0	6.8E-02	1.02	0.0E-02	-6.30E-04	-6.30E-04	
353.0	6.9E-02	1.02	0.0E-02	-6.41E-04	-6.41E-04	
359.0	7.0E-02	1.02	0.0E-02	-6.52E-04	-6.52E-04	
365.0	7.1E-02	1.02	0.0E-02	-6.63E-04	-6.63E-04	
371.0	7.2E-02	1.02	0.0E-02	-6.74E-04	-6.74E-04	
377.0	7.3E-02	1.02	0.0E-02	-6.85E-04	-6.85E-04	
383.0	7.4E-02	1.02	0.0E-02	-6.96E-04	-6.96E-04	
389.0	7.5E-02	1.02	0.0E-02	-7.07E-04	-7.07E-04	
395.0	7.6E-02	1.02	0.0E-02	-7.18E-04	-7.18E-04	
401.0	7.7E-02	1.02	0.0E-02	-7.29E-04	-7.29E-04	
407.0	7.8E-02	1.02	0.0E-02	-7.40E-04	-7.40E-04	
413.0	7.9E-02	1.02	0.0E-02	-7.51E-04	-7.51E-04	
419.0	8.0E-02	1.02	0.0E-02	-7.62E-04	-7.62E-04	
425.0	8.1E-02	1.02	0.0E-02	-7.73E-04	-7.73E-04	
431.0	8.2E-02	1.02	0.0E-02	-7.84E-04	-7.84E-04	
437.0	8.3E-02	1.02	0.0E-02	-7.95E-04	-7.95E-04	
443.0	8.4E-02	1.02	0.0E-02	-8.06E-04	-8.06E-04	
449.0	8.5E-02	1.02	0.0E-02	-8.17E-04	-8.17E-04	
455.0	8.6E-02	1.02	0.0E-02	-8.28E-04	-8.28E-04	
461.0	8.7E-02	1.02	0.0E-02	-8.39E-04	-8.39E-04	
467.0	8.8E-02	1.02	0.0E-02	-8.50E-04	-8.50E-04	
473.0	8.9E-02	1.02	0.0E-02	-8.61E-04	-8.61E-04	
479.0	9.0E-02	1.02	0.0E-02	-8.72E-04	-8.72E-04	
485.0	9.1E-02	1.02	0.0E-02	-8.83E-04	-8.83E-04	
491.0	9.2E-02	1.02	0.0E-02	-8.94E-04	-8.94E-04	
497.0	9.3E-02	1.02	0.0E-02	-9.05E-04	-9.05E-04	
503.0	9.4E-02	1.02	0.0E-02	-9.16E-04	-9.16E-04	
509.0	9.5E-02	1.02	0.0E-02	-9.27E-04	-9.27E-04	
515.0	9.6E-02	1.02	0.0E-02	-9.38E-04	-9.38E-04	
521.0	9.7E-02	1.02	0.0E-02	-9.49E-04	-9.49E-04	
527.0	9.8E-02	1.02	0.0E-02	-9.60E-04	-9.60E-04	
533.0	9.9E-02	1.02	0.0E-02	-9.71E-04	-9.71E-04	
539.0	1.00E-01	1.02	0.0E-02	-9.82E-04	-9.82E-04	
545.0	1.01E-01	1.02	0.0E-02	-9.93E-04	-9.93E-04	
551.0	1.02E-01	1.02	0.0E-02	-1.004E-03	-1.004E-03	
557.0	1.03E-01	1.02	0.0E-02	-1.015E-03	-1.015E-03	
563.0	1.04E-01	1.02	0.0E-02	-1.026E-03	-1.026E-03	
569.0	1.05E-01	1.02	0.0E-02	-1.037E-03	-1.037E-03	
575.0	1.06E-01	1.02	0.0E-02	-1.048E-03	-1.048E-03	
581.0	1.07E-01	1.02	0.0E-02	-1.059E-03	-1.059E-03	
587.0	1.08E-01	1.02	0.0E-02	-1.070E-03	-1.070E-03	
593.0	1.09E-01	1.02	0.0E-02	-1.081E-03	-1.081E-03	
599.0	1.10E-01	1.02	0.0E-02	-1.092E-03	-1.092E-03	
605.0	1.11E-01	1.02	0.0E-02	-1.103E-03	-1.103E-03	
611.0	1.12E-01	1.02	0.0E-02	-1.114E-03	-1.114E-03	
617.0	1.13E-01	1.02	0.0E-02	-1.125E-03	-1.125E-03	
623.0	1.14E-01	1.02	0.0E-02	-1.136E-03	-1.136E-03	
629.0	1.15E-01	1.02	0.0E-02	-1.147E-03	-1.147E-03	
635.0	1.16E-01	1.02	0.0E-02	-1.158E-03	-1.158E-03	
641.0	1.17E-01	1.02	0.0E-02	-1.169E-03	-1.169E-03	
647.0	1.18E-01	1.02	0.0E-02	-1.180E-03	-1.180E-03	
653.0	1.19E-01	1.02	0.0E-02	-1.191E-03	-1.191E-03	
659.0	1.20E-01	1.02	0.0E-02	-1.202E-03	-1.202E-03	
665.0	1.21E-01	1.02	0.0E-02	-1.213E-03	-1.213E-03	
671.0	1.22E-01	1.02	0.0E-02	-1.224E-03	-1.224E-03	
677.0	1.23E-01	1.02	0.0E-02	-1.235E-03	-1.235E-03	
683.0	1.24E-01	1.02	0.0E-02	-1.246E-03	-1.246E-03	
689.0	1.25E-01	1.02	0.0E-02	-1.257E-03	-1.257E-03	
695.0	1.26E-01	1.02	0.0E-02	-1.268E-03	-1.268E-03	
701.0	1.27E-01	1.02	0.0E-02	-1.279E-03	-1.279E-03	
707.0	1.28E-01	1.02	0.0E-02	-1.290E-03	-1.290E-03	
713.0	1.29E-01	1.02	0.0E-02	-1.301E-03	-1.301E-03	
719.0	1.30E-01	1.02	0.0E-02	-1.312E-03	-1.312E-03	
725.0	1.31E-01	1.02	0.0E-02	-1.323E-03	-1.323E-03	
731.0	1.32E-01	1.02	0.0E-02	-1.334E-03	-1.334E-03	
737.0	1.33E-01	1.02	0.0E-02	-1.345E-03	-1.345E-03	
743.0	1.34E-01	1.02	0.0E-02	-1.356E-03	-1.356E-03	
749.0	1.35E-01	1.02	0.0E-02	-1.367E-03	-1.367E-03	
755.0	1.36E-01	1.02	0.0E-02	-1.378E-03	-1.378E-03	
761.0	1.37E-01	1.02	0.0E-02	-1.389E-03	-1.389E-03	
767.0	1.38E-01	1.02	0.0E-02	-1.399E-03	-1.399E-03	
773.0	1.39E-01	1.02	0.0E-02	-1.410E-03	-1.410E-03	
779.0	1.40E-01	1.02	0.0E-02	-1.421E-03	-1.421E-03	
785.0	1.41E-01	1.02	0.0E-02	-1.432E-03		

TABLE III

CROSS-SPECTRUM STATISTICS FOR 70038
 BETWEEN CHANNELS 101SA0 UAC SONIC TAPE 140/1/10/9/68
 AND 2SONIC W TAPE 140/1/10/9/68
 STATISTICS ARE BASED ON 102 BLOCKS OF 1024 SAMPLES EACH
 THE SAMPLING FREQUENCY WAS 62.77105AMP/SEC
 MAKING THE BLOCK LENGTH 16.31326 SECONDS
 TREND IS THE AVERAGE OF
 $(\text{VALUE(A)} - \text{VALUE(B)}) / (\text{BLOCK NO. (A)} - \text{BLOCK NO. (B)})$
 PHASE POSITIVE MEANS SECOND CHANNEL LAGS FIRST
 CALIBRATION FACTORS OF -8.05E-01 AND
 6.540E-01 HAVE BEEN APPLIED TO THE INPUT DATA

	FREQUENCY HERTZ	CO-SPECTRUM 1/M/SEC	STD. DEV. *M/SEC	TREND /HERTZ
1	5.31E-02	-1.33E-01	2.49E-01	1.56E-03
2	1.19E-01	-8.27E-02	1.26E-01	1.60E-04
3	1.81E-01	-6.52E-02	1.15E-01	2.68E-04
4	2.43E-01	-4.78E-02	7.76E-02	2.89E-04
5	3.32E-01	-2.84E-02	4.19E-02	-3.55E-05
6	4.56E-01	-2.50E-02	3.38E-02	5.88E-05
7	6.06E-01	-1.15E-02	1.95E-02	7.14E-05
8	8.18E-01	-1.08E-02	1.44E-02	2.80E-05
9	1.09E-00	-7.32E-03	9.14E-03	5.33E-05
10	1.46E-00	-2.91E-03	4.87E-03	2.61E-05
11	1.94E-00	-2.32E-03	3.01E-03	1.38E-05
12	2.58E-00	-9.55E-04	1.78E-03	6.44E-06
13	3.43E-00	-6.73E-04	8.97E-04	3.43E-06
14	4.58E-00	-4.49E-04	5.30E-04	9.28E-07
15	6.13E-00	-2.48E-04	2.50E-04	1.01E-06
16	8.16E-00	-1.27E-04	1.32E-04	1.00E-06
17	1.09E-01	-4.29E-05	7.32E-05	2.46E-07
18	1.45E-01	4.56E-06	2.27E-05	-8.67E-08
19	1.94E-01	5.76E-06	1.17E-05	-6.49E-08
20	2.58E-01	-2.92E-07	4.41E-06	-2.25E-08
21	3.16E-01	-7.10E-07	4.43E-06	-6.87E-09

INTEGRAL (SUM) UNDER COSPECTRUM = -3.25E-02 M/SEC

TABLE IV

Integral (Sum) under Quadspectrum = -6.21E-03 M/sec

TABLE V

Fourier Coefficients First 114 Components

COEF	FREQ	AMPL	Coeff										
1	0.05	3.64E-01	0.95	0.64E-01	0.92	0.64E-01	0.90	0.64E-01	0.88	0.64E-01	0.86	0.64E-01	0.83
2	0.12	2.41E-01	0.69	2.2E-02	0.69								
3	0.18	1.65E-01	0.69	1.8E-02	0.69								
4	0.25	1.45E-01	0.56	1.5E-02	0.56								
5	0.31	1.32E-01	0.56	1.2E-02	0.56								
6	0.37	1.02E-01	0.56	9.4E-03	0.56								
7	0.43	1.02E-01	0.56	7.6E-03	0.56								
8	0.49	1.02E-01	0.56	6.0E-03	0.56								
9	0.55	8.0E-02	0.56	4.5E-03	0.56								
10	0.61	7.3E-02	0.56	3.2E-03	0.56								
11	0.67	7.0E-02	0.56	3.0E-03	0.56								
12	0.74	6.0E-02	0.56	3.0E-03	0.56								
13	0.80	6.0E-02	0.56	3.0E-03	0.56								
14	0.86	6.0E-02	0.56	3.0E-03	0.56								
15	0.92	5.0E-02	0.56	4.2E-02	0.56								
16	0.98	5.0E-02	0.56	6.4E-02	0.56								
17	1.04	5.0E-02	0.56	6.6E-02	0.56								
18	1.10	5.0E-02	0.56	3.4E-02	0.56								
19	1.16	5.0E-02	0.56	7.4E-02	0.56								
20	1.23	4.0E-02	0.56	9.8E-02	0.56								
21	1.29	4.0E-02	0.56	1.4E-01	0.56								
22	1.35	3.0E-02	0.56	1.5E-01	0.56								
23	1.41	4.0E-02	0.56	1.2E-01	0.56								
24	1.47	4.0E-02	0.56	7.4E-02	0.56								
25	1.53	4.0E-02	0.56	3.1E-01	0.56								
26	1.59	4.0E-02	0.56	8.5E-02	0.56								
27	1.66	3.0E-02	0.56	2.0E-01	0.56								
28	1.72	3.0E-02	0.56	1.4E-01	0.56								
29	1.78	3.0E-02	0.56	6.3E-02	0.56								
30	1.84	3.0E-02	0.56	2.0E-01	0.56								
31	1.90	3.0E-02	0.56	1.4E-01	0.56								
32	1.96	2.0E-02	0.56	1.4E-01	0.56								
33	2.02	2.0E-02	0.56	1.2E-01	0.56								
34	2.08	2.0E-02	0.56	7.2E-02	0.56								
35	2.14	2.0E-02	0.56	5.9E-02	0.56								
36	2.21	2.0E-02	0.56	8.3E-02	0.56								
37	2.27	2.0E-02	0.56	1.1E-01	0.56								
38	2.33	2.0E-02	0.56	7.2E-02	0.56								
39	2.45	2.0E-02	0.56	2.0E-01	0.56								
40	2.51	2.0E-02	0.56	7.1E-02	0.56								
41	2.57	2.0E-02	0.56	4.3E-01	0.56								
42	2.64	2.0E-02	0.56	4.8E-01	0.56								
43	2.70	2.0E-02	0.56	6.7E-01	0.56								
44	2.76	2.0E-02	0.56	5.3E-01	0.56								
45	2.82	2.0E-02	0.56	3.5E-01	0.56								
46	2.88	2.0E-02	0.56	4.4E-01	0.56								
47	2.94	2.0E-02	0.56	2.8E-01	0.56								
48	3.00	2.0E-02	0.56	1.3E-01	0.56								
49	3.06	2.0E-02	0.56	1.9E-01	0.56								
50	3.12	2.0E-02	0.56	2.5E-01	0.56								
51	3.18	2.0E-02	0.56	1.9E-01	0.56								
52	3.24	2.0E-02	0.56	2.1E-01	0.56								
53	3.30	2.0E-02	0.56	1.9E-01	0.56								
54	3.36	2.0E-02	0.56	2.1E-01	0.56								
55	3.42	2.0E-02	0.56	1.9E-01	0.56								
56	3.48	2.0E-02	0.56	2.1E-01	0.56								
57	3.54	2.0E-02	0.56	1.9E-01	0.56								

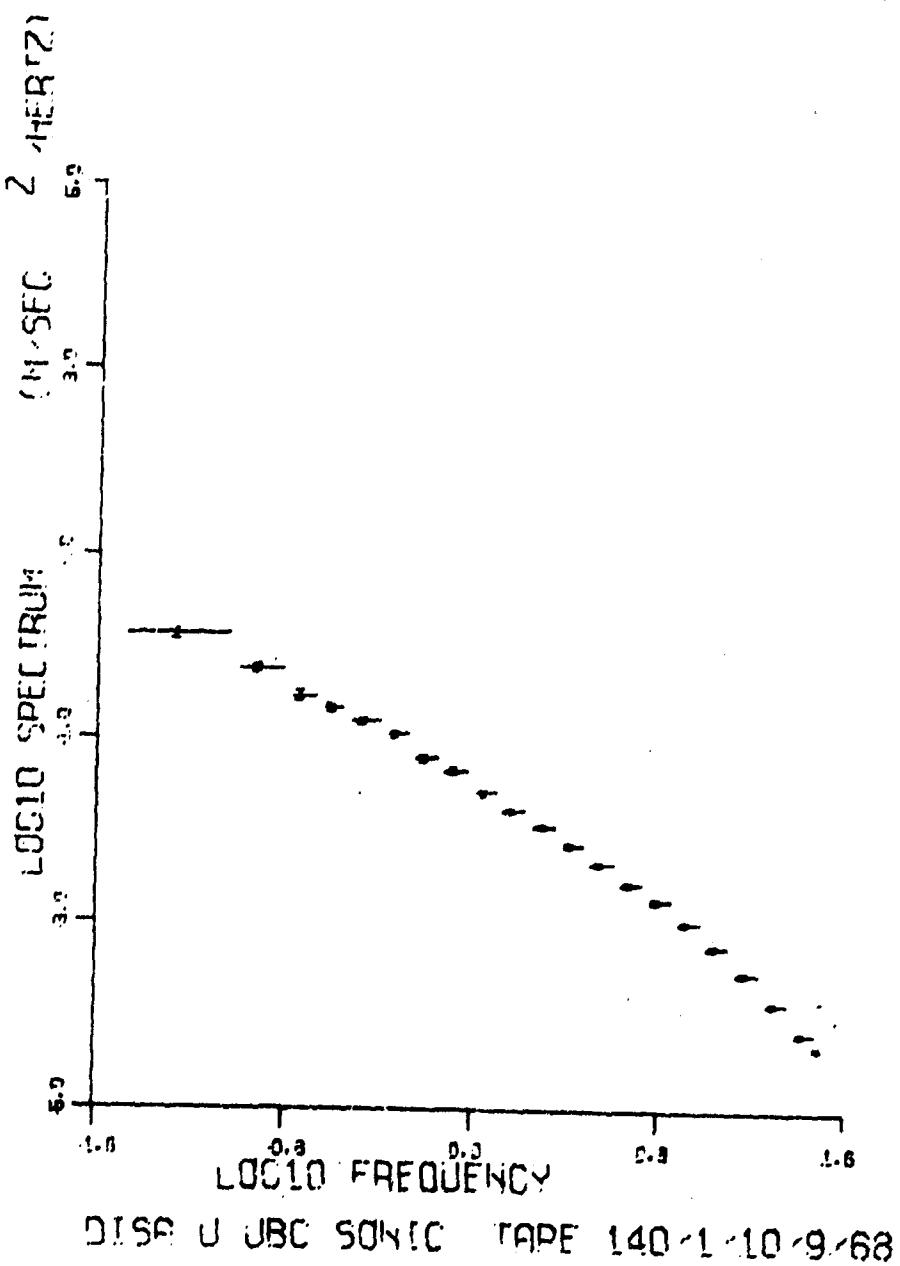


FIGURE 2. DISA U SPECTRUM

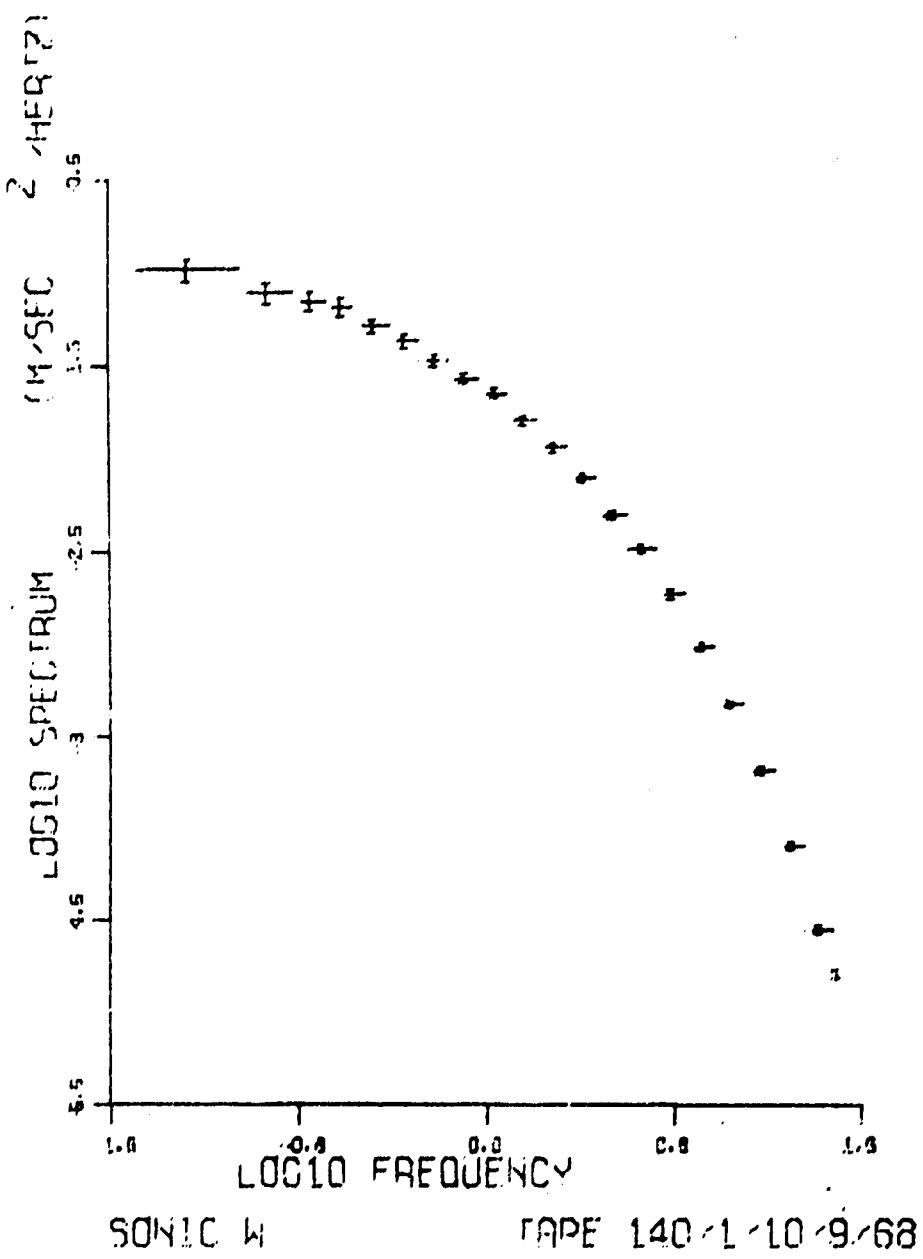


FIGURE 3. SONIC W SPECTRUM

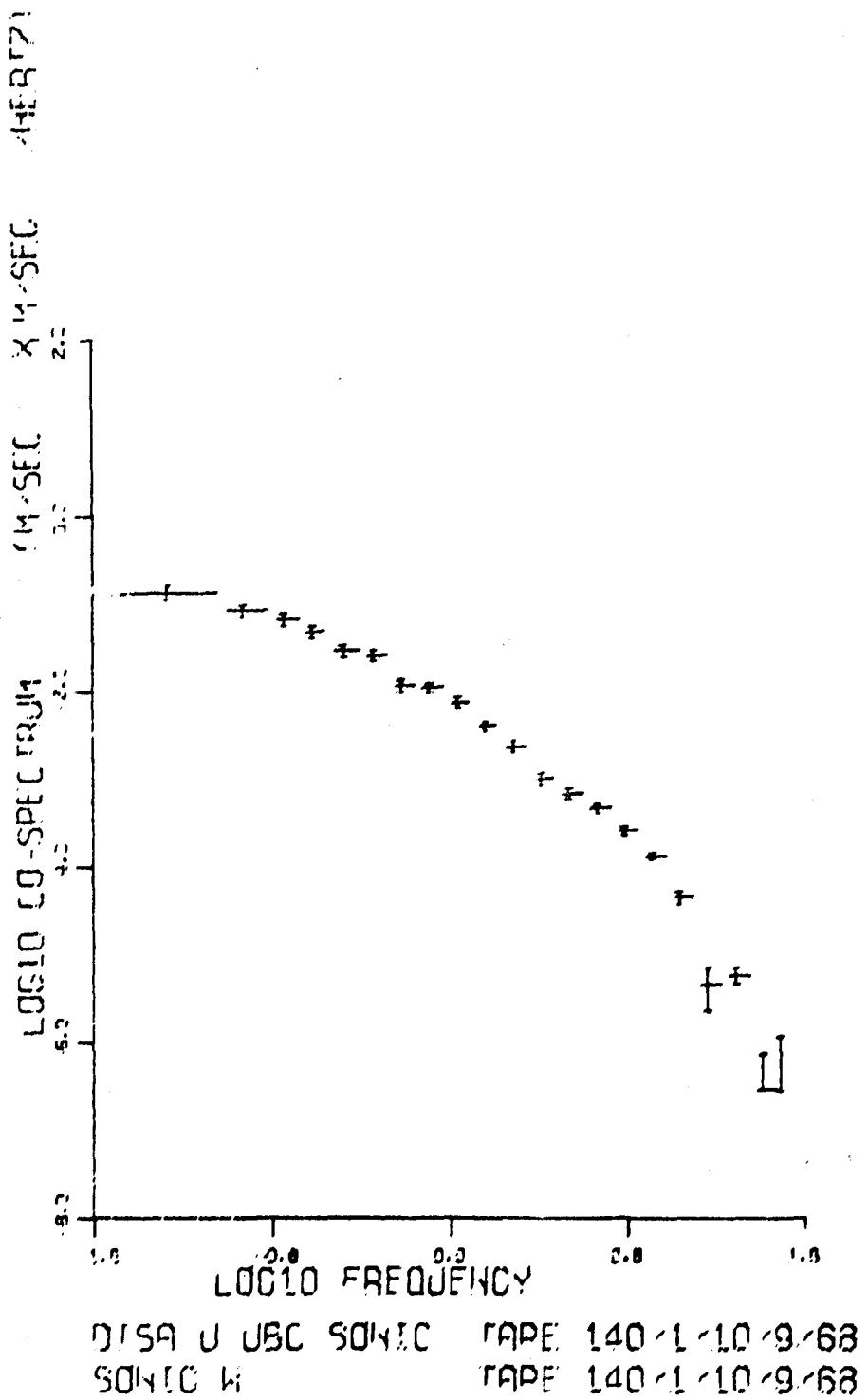


FIGURE 4. SONIC W, DISA U CO-SPECTRUM

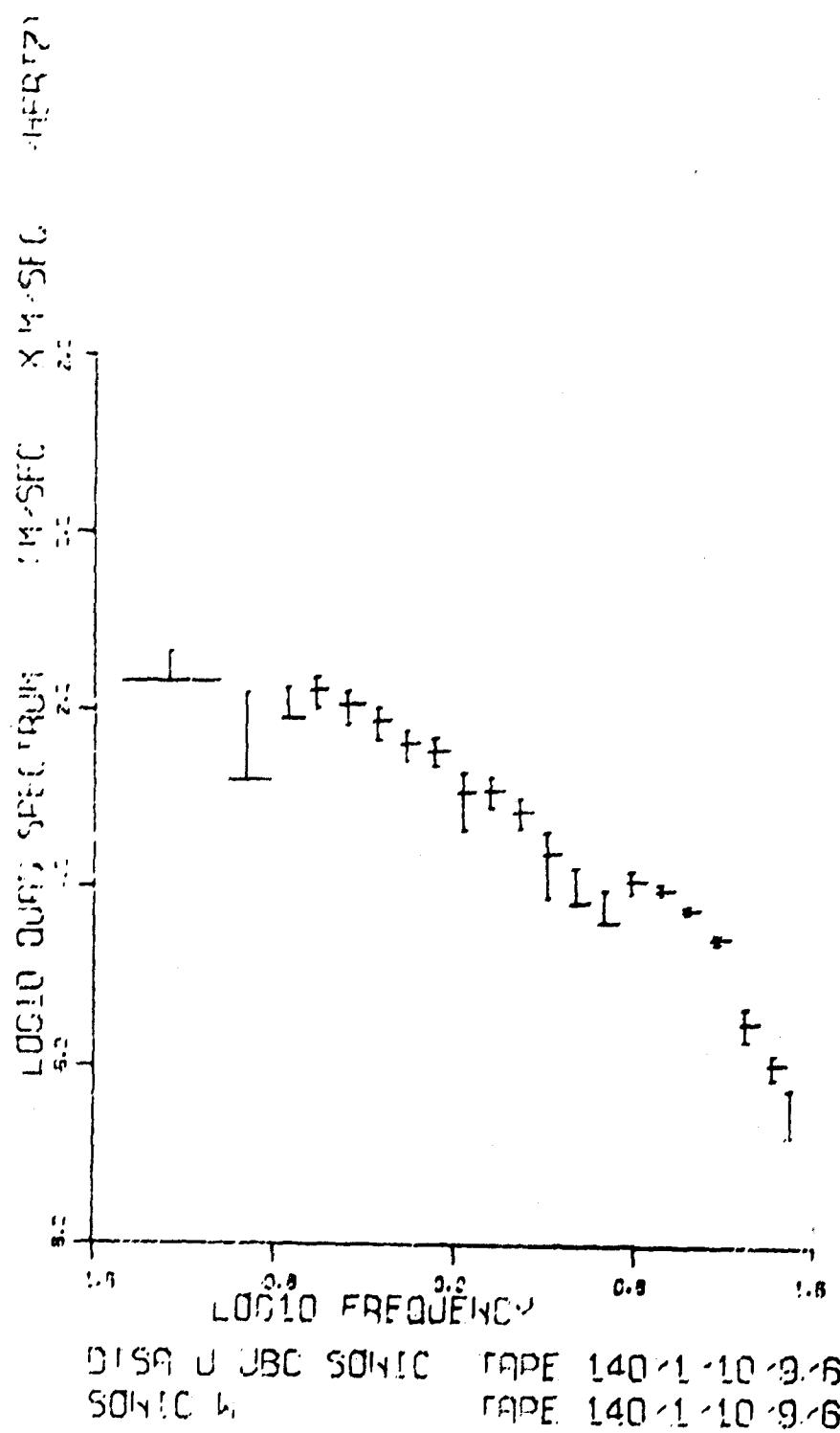


FIGURE 5. SONIC W, DISA U QUAD-SPECTRUM

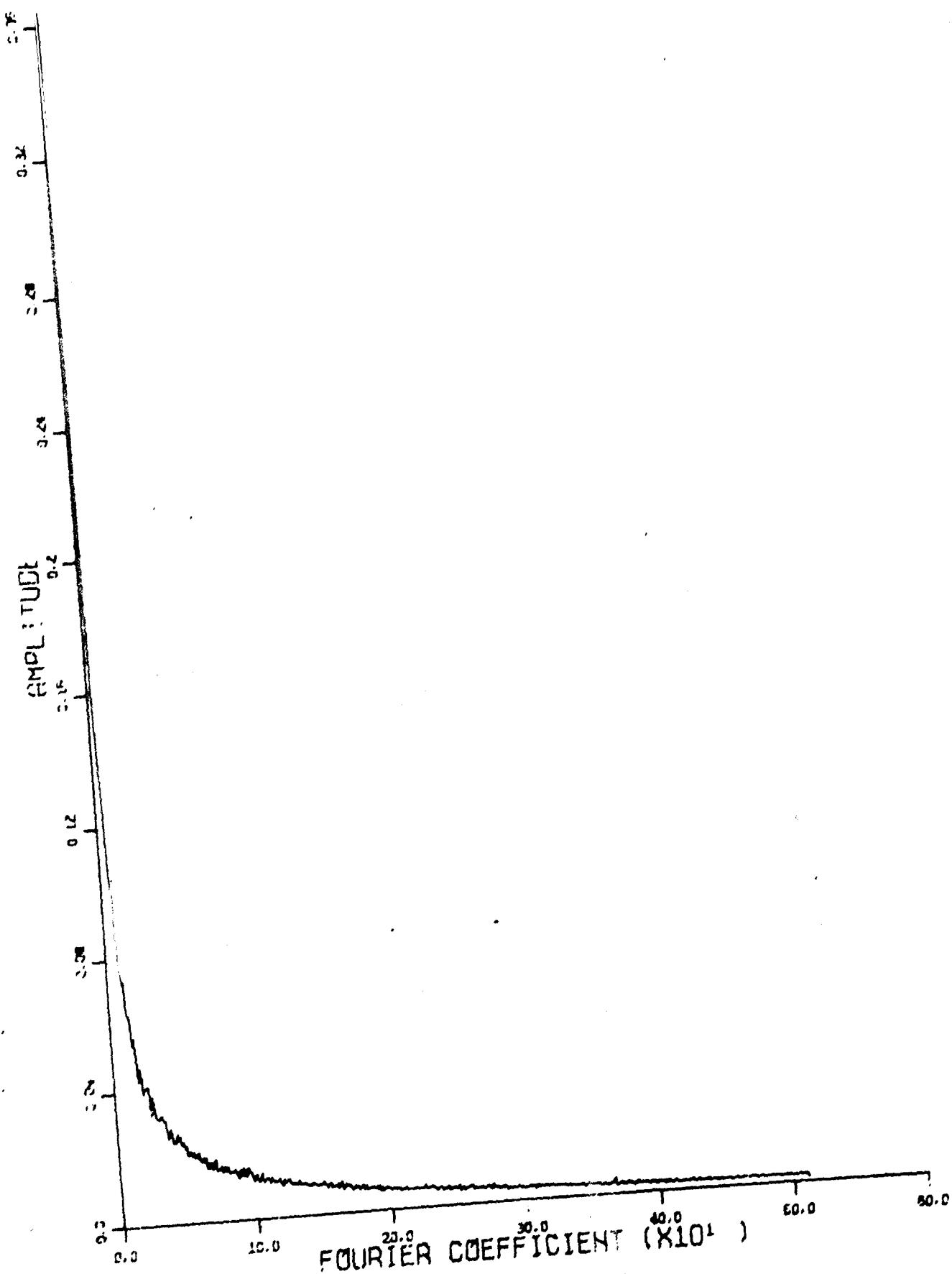


FIGURE 6. DISA U FOURIER COEFFICIENTS

IV INPUT DATA TAPE FORMAT

The input data tape is read at the Fortran level by an unformatted read. The read is in the subroutine OCEAN 1 and has the form:

```
READ (NUNFT) JCTR, JCHANS, (DATA(J), J = 1, JCTR)
```

where JCHANS is the number of channels digitized and JCTR is the number of data words in the tape record. DATA is an array into which the data is to be read and is single precision real and dimensioned 256.

To create a suitable tape with 4 channels of information the programmer would set JCHANS = 4 then line up floating point data words in a single dimensioned array, DATA, in the order:

Cross	Channel 1	DATA(1)
Channel	Channel 2	DATA(2)
Sweep 1	Channel 3	" (3)
	Channel 4	" (4)
	Channel 1	" (5)
Cross	Channel 2	" (6)
Channel	Channel 3	" (7)
Sweep 2	Channel 4	" (8)
	Channel 1	" (9)
.	.	.
.	.	.
.	.	.
	Channel 1	.
Cross	Channel 2	.
Channel	Channel 3	.
Sweep K	Channel 4	DATA(JCTR)

where

JCTR = 4 * K and is less than 256.

The statement

WRITE(KUNFT) JCTR, JCHANS, (DATA(J), J = 1, JCTR)

is then executed.

This loading of the array and writing it out is repeated until all data has been transferred to tape and then an end of file is written. It is not necessary that JCTR be the same for all records written and it may be zero. It must be however an integral number of cross channel sweeps.

V PROCEDURE IN ANALYZING DATA

The first requirement is to produce a tape containing the data in the format that has been described. This tape is used as input to the FTOR program and a second tape is produced containing the Fourier coefficients. This tape is then used with either the SCOR or FCPILOT program depending on the analysis desired. If co-spectra or quadrature spectra are desired then SCOR must be used. If power spectra alone are desired SCOR or FCPILOT can be used. If more detail than SCOR supplies for the spectra, or a calcomp or printed plot of the individual Fourier coefficient amplitudes are desired, FCPILOT should be used.

The format of the deck of control cards required to run each of the programs is described in detail on the comments cards with the Fortran source decks (APPENDIX)

The programs were originally written for an IBM 7044 (Garrett, 1967) and were then modified for an IBM 360/67 running under the Michigan Terminal System. Neither of these systems have the set of utilities for the tape handling available under OS/MUT. For this reason all tape positioning on files is handled by the programs at the Fortran level. It is therefore necessary to have a DD card present in the deck for each file on the tape whether it is read or written or just spaced over.

VI SOURCE PROGRAMS

The system consists of three programs:

1. UBC FTOR
2. UBC SCOR - 150K in FORT step to compile.
3. UBC FCPILOT

The source programs have been written as files 1, 2 and 3 in the order above on the tape NPS216. The DSNAME's are UBCFTOR, UBCSCOR and UBCFCPLT. The DCB is (RECFM = FB, LRECL = 80, BLKSIZE = 1600).

The object programs are on disc pack FAC001 in load module form. The DSNAME for the file is F1178.TSLIB.

UBCFTOR

UBCSCOR

UBCFCPLT

ACKNOWLEDGEMENTS

This work was supported by the Naval Ordnance Systems Command under NAVORD Contract ORD TASK-036-005/561-1/VR104-03-01.

The time series program was set up and tested on the IBM 360/67 computer of the Naval Postgraduate School by Mr. Ron Wilson. We wish to thank the Institute of Oceanography of the University of British Columbia for making Mr. Wilson and the programs available to us.

Professor Doug Williams, Director of the Computing Facility of the Naval Postgraduate School, and his staff assisted us greatly in seeing that this work went smoothly.

REFERENCES

- Blackman, R. B. and J. W. Tukey, 1958. The Measurement of Power Spectra, Dover Press, New York.
- Cooley, J. W. and J. W. Tukey, 1965. "An Algorithm for the Machine Calculation of Complex Fourier Series". Mathematics of Computation, 19 (90), 297-301.
- Garrett, John. Interim Report on Digital Data Analysis. December, 1967. Internally distributed report. Institute of Oceanography, U.B.C. Canada.

APPENDIX 1

ANALYZING DATA DIGITIZED ON THE SDS-9300

Two computer programs are required to develop the capability to use the SDS-9300 and the associated analog computer to digitize data to be analyzed by the time series programs.

The first is an SDS-9300 program to control the digitization procedure and produce a 7-track magnetic tape containing the digital data. The second program is an IBM 360 program to convert the 7-track tape to a 9-track tape in the format described in this report for the input tape to the time series program.

The SDS-9300 program should do approximately the following:

1. It should allow for manual starting and stopping of the digitization process by the computer operator when provided with suitable signals (audio tones, oscilloscope signal changes, verbal instructions from the person concerned with the analog signal, etc.)
2. It should allow for a variable number of analog input channels. This number should be an input to the program at execution time. Ten would be a safe maximum since the time series system of programs can handle only ten.
3. The program should be written so that when the interrupt is given all channels are sampled as rapidly as possible and the data accumulated for tape output. Only after a cross channel sequence is completed should a test and possible tape output occur if enough data has been accumulated.
4. Care should be taken to make the program efficient in the sense that the computer is always ready to service the clock interrupts

when they occur. Further, care should be taken to make the records long enough that the tape is used efficiently.

5. The program for the IBM 360/67 should take this 7-track tape and produce a 9-track tape in the format required by the time series programs.
6. It is anticipated that multifile tapes will be necessary to avoid piling up large numbers of magnetic tapes with only the first part of the tape used. Therefore these two programs should be made capable of positioning themselves in multifile tapes under the control of either the job control language or input control cards to the program.

APPENDIX II
FORTRAN SOURCE PROGRAMS

FOURIER TRANSFORM OF TIME SERIES DATA SUPPLIED THROUGH
'OCEAN' SUBROUTINES USING P-K FORT FAST FOURIER TRANSFORM
SUBROUTINE (SHARE SDA3465) IN THE IBM SYSTEM 360 MODEL 67.

LAST REVISION JANUARY 24, 1969

JOHN GARRETT

NBLOCK BLOCKS OF $2^{**}NPOW$ SAMPLES EACH ARE READ FOR KCHAN OF THE
NCHAN CHANNELS AVAILABLE TO THE 'OCEAN' SUBROUTINES. THE CONTENTS OF
ANY CHANNEL MAY BE REPLACED BY A LINEAR COMBINATION OF ITSELF WITH ANY
OTHER CHANNEL.

FOR EACH BLOCK $2^{**}(NPOW-1)$ COMPLEX FOURIER COEFFICIENTS ARE
COMPUTED FOR EACH OF THE KCHAN CHANNELS. THESE ARE THEN WRITTEN IN THE
OUTPUT (TAPE) 03 IN THE FORMAT DESCRIBED BELOW. IN ADDITION THE
COEFFICIENTS MAY BE SUMMED IN GROUPS OF $(2^{**}(NPOW-1))/32$ AND
PRINTED OUT.

ADDITIONAL OPTIONS ARE DESCRIBED UNDER THE RELEVANT CONTROL
PARAMETERS BELOW.

IT SHOULD BE NOTED THAT THE COEFFICIENTS PRODUCED ARE THOSE OF THE
FOURIER SERIES

$y(j) = \sum_{k=0, n/2}^n$ OF REAL PARTS OF
 $(c(k)*\exp((2*\pi*i/n)*j*k))$
WITH $j = 0, n-1, y(j)$ REAL, AND $i = \sqrt{-1}$

THE FOLLOWING SUBROUTINES ARE REQUIRED
OCEAN1, OCEAN2, OCEAN3, RWUNL0
SKPFL
CONVCL
USCRMB
P-K FORT

THE FOLLOWING LOGICAL INPUT/OUTPUT DEVICES ARE USED IN THIS PROGRAM
2 = SCRATCH TAPE FOR TEMPORARY STORAGE OF COEFFICIENTS IF
LOFR=1 BELOW
3 = OUTPUT (TAPE) FOR COEFFICIENTS
5 = (CARD) INPUT FOR CONTROL PARAMETERS
6 = PRINTED OUTPUT
INUNIT = INPUT TAPE OF TIME SERIES DATA FOR 'OCEAN' SUBROUTINES

INPUT INFORMATION REQUIRED

FIRST DATA CARD IN COLUMN NUMBER
1-9 IDUSER = USER IDENTIFICATION NUMBER (9-DIGIT INTEGER)
14-15 NCHAN = NUMBER OF CHANNELS DIGITIZED ON OCEAN TAPE
22 NTYPE = (NOT RELEVANT. SET TO ZERO.)
34-35 INFILE = FILE NUMBER OF DATA ON OCEAN TAPE

44-45 INUNIT = NUMBER OF UNIT ON WHICH INPUT TAPE IS MOUNTED
55 NSEARCH = (NOT RELEVANT. SET TO ZERO.)
61-70 SAMFRQ = SAMPLING FREQUENCY OF DIGITIZING, (SAMPLES/SECOND)
(MUST INCLUDE A DECIMAL POINT)

SECOND DATA CARD

4-5 NBLOCK = NUMBER OF BLOCKS DESIRED
14-15 NPOW-- MAXIMUM NUMBERS OF SAMPLES PER BLOCK WILL BE 2^{**NPOW} ,
MAX NPOW IS 13 (8192 SAMPLES/BLOCK) BUT NPOW WILL BE
REDUCED UNTIL $(2^{**NPOW}) * KCHAN$ WILL FIT IN MEMORY. FOR
KCHAN OF 10 THIS WILL GIVE NPOW = 10 (1024 SAMPLES/BLOCK).
24-25 MTAPE = +1 FOR OUTPUT TAPE TO BE WRITTEN
= -1 FOR NO OUTPUT TAPE
34-35 NFILE - OUTPUT COEFFICIENTS WILL BE NFILE-TH FILE ON TAPE
43-45 MAXERR = (NOT RELEVANT. SET TO ZERO.)
54-55 MPRINT = -1 SUPPRESSES SUMMARY COEFFICIENTS PRINT OUT
= 0 OR GREATER PERMITS PRINT OUT

THIRD DATA CARD

4-5 KCHAN = NUMBER OF CHANNELS TO BE TRANSFORMED (MAX 10)
15 LOFR = 2 IF COEFFICIENTS TO BE COMPUTED FROM DATA SMOOTHED
AND SUBSAMPLED (DECIMATED) USING CONVOL SUBROUTINE.
WEIGHTS USED IN SMOOTHING AND DECIMATING FACTOR ARE
DETERMINED BY CHOICE OF CONVOL USED.
= 1 IF ALTERNATE BLOCKS TO BE MADE UP OF SAMPLES FROM DATA
SMOOTHED AND DECIMATED USING CONVOL SUBROUTINE.
COEFFICIENTS WILL APPEAR ON OUTPUT TAPE IN FILE
IMMEDIATELY FOLLOWING THAT CONTAINING RESULTS FROM
UNSMOOTHED DATA
25 IHANN = 1 IF FOURIER COEFFICIENTS TO BE HANNED AND
NORMALIZED (*SQRT(8/3))
= 0 IF NOT

NEXT KCHAN CARDS

1-5 NO. OF PRIMARY A TO D CHANNEL
6-10 NO. OF SECONDARY A TO D CHANNEL
11-20 CALIBRATION ASSOCIATED WITH THE PRIMARY A TO D CHANNEL
21-30 CALIBRATION ASSOCIATED WITH THE SECONDARY A TO D CHANNEL
31-66 ALPHAMERIC NAME OF RESULTING PRIMARY CHANNEL
71-78 8 CHARACTER NAME OF UNITS FOR RESULTING PRIMARY CHANNEL

THE DATA TRANSFORMED AS THE PRIMARY CHANNEL IS THEN

CALIBRATION1 X VALUE OF PRIMARY + CALIBRATION2 X VALUE OF
SECONDARY CHANNEL

C A SECOND SET OF DATA CARDS WILL PRODUCE A SECOND ANALYSIS. A BLANK
C CARD TERMINATES THE RUN.

C FORMAT OF OUTPUT TAPE FOR EACH BLOCK TRANSFORMED IS AS FOLLOWS
FIRST LOGICAL RECORD IS AN ARRAY OF 256 WORDS CALLED

ARTAPE(4) = NTAP (SEE BELOW)

ARTAPE(1) = IDUSER

ARTAPE

ARTAPE(3) = NUMBER OF SAMPLES /BLOCK

ARTAPE(2) = BLOCK NUMBER

ARTAPE(10) = IHANN

ARTAPE(5) = NUMBER OF CHANNELS TRANSFORMED

ARTAPE(6) = SAMPLING FREQUENCY

ARTAPE(10+K) = MCHAN(K)

ARTAPE(31+9(K-1)) = ACHNAM(K) (9A4)

ARTAPE(121+2(K-1)) = AUNITS(1,K) (2A4)

ARTAPE(140+K) = CAL(K)

NEXT NTAP LOGICAL RECORDS OF 256 WORDS EACH CALLED
TAPRAY AND CONSIST OF

1 WORD CONTAINING INTEGER HARMONIC NO (0 TO IBLOCK/2) FOLLOWED BY
2*KCHAN WORDS CONTAINING FOR EACH OF THE KCHAN CHANNELS THE
REAL PART OF THE FOURIER COEFF AT THIS HARMONIC (FIRST WORD)
FOLLOWED BY THE IMAGINARY PART (SECOND WORD).

ONLY COMPLETE SEQUENCES OF 1+(2*KCHAN) WORDS ARE INCLUDED IN A
TAPRAY SO THAT THE LAST FEW WORDS MAY CONTAIN ZEROS

AN END OF FILE IS WRITTEN ON THE OUTPUT TAPE AT THE END OF A SEQUENCE
OF BLOCKS

J.F. GARRETT
J.P. WILSON

INSTITUTE OF OCEANOGRAPHY
UNIVERSITY OF BRITISH COLUMBIA
VANCOUVER 8, CANADA

```
DIMENSION W(10242), S(4100), ARTAPE(256), TAPRAY(256), MCHAN(10),
1 CAL(10), AUNITS(2,10), ACHNAM(9,10), INOCH(12), I2TAB(14), INCR(10),
2 INDATA(12), W1(1024), W2(1024), W3(1024), W4(1024), W5(1024),
3 W6(1024), W7(1024), W8(1024), W9(1024), W10(1024), A(256), ATA(12)
4 , VAL(50,10)
DIMENSION MCHAN2(10), CAL2(10), DDFREQ(20)
COMMON V$, NAN, KOUNT
EQUIVAL 'CE' (W(1), W1(1)), (W(1025), W2(1)), (W(2049), W3(1)),
1 (W(30), W4(1)), (W(4097), W5(1)), (W(5121), W6(1)), (W(6145), W7(1)),
2 (W(71), W8(1)), (W(8193), W9(1)), (W(9217), W10(1)), (ARTAPE(1), A(1)),
3, (A(1), IDUSER), (A(2), MBLOCK), (A(3), IBLOCK), (A(4), NTAP),
4, (A(5), KCHAN), (A(6), SAMFRQ), (A(11), MCHAN(1)),
5, (A(12), ACHNAM(1,1)), (A(121), AUNITS(1,1)), (A(141), CAL(1)),
6, (A(151), INOCH(1)), (A(163), I2TAB(1)), (A(180), INCR(1)),
7, (A(190), INDATA(1)), (A(210), TAPRAY(1)), (A(210), IAR),
8, (A(190), ATA(1))
```

```

NZBLCK = 0
CORNB = 1./SQRT(6.0)
CURNA = 2.0 * CORNB
ICFPCS = 0
6000 READ (5,5,END=6010) IDUSER,NCHAN,NTYPE,INFILE,INUNIT,NSEARH,SAMFRQ
IF (INFILE.EQ.0) GO TO 6010
CFREQ = SAMFRQ
DO 10010 I=1,10
MCHAN(I) = 0
MCHAN2(I)=0
CAL(I)=0
CAL2(I)=0.
DO 10009 J=1,9
10009 ACHNAM(J,I) = 0
DO 10010 J=1,2
AUNITS(J,I) = 0
10010 CONTINUE
DO 10012 I=1,12
10012 INDCH(I) = 0
READ (5,6) NBLOCK,NPOW,MTAPE,KFILE,MAXERR,MPRINT,IRWCTL
IF (ICFPCS.EQ.0) NFILE = KFILE
READ(5,6) KCHAN ,LOFR,IHANN
ARTAPF(10) = IHANN
IF (LCFR.EQ.1) REWIND 2
DO 8 K = 1,KCHAN
READ (5,7) MCHAN(K),MCHAN2(K),CAL(K),CAL2(K),(ACHNAM(L,K),L=1,9),
1(AUNITS(L,K),L=1,2)
IF (MCHAN2(K).EQ.0) MCHAN2(K) = 1
IF (MCHAN(K).EQ.0) GO TO 9
KCH = MCHAN(K)
INDCH(KCH) = K
8 KCH = K
9 KCHAN = KCH
C SET UP TWOS TABLE
DO 10 N =1,14
10 I2TAB(N) = 2**N
C DETERMINE BLOCK SIZE
11 LOKTES = KCHAN*I2TAB(NPOW)
IF(LOKTES.LE.10242) GO TO 15
NPOW = NPCW-1
GO TO 11
15 IBLOCK = I2TAB(NPOW)
KBLOCK = IBLOCK/2
KCH = 256/(I+I2*KCHAN)
NTAP =(KBLOCK + KCH -1)/KCH
C COMPUTE SINE TABLE
CALL FORT(W,NPOW,S,O,IFERR)
C POSITION OUTPUT TAPE

```

```

IF (MTAPE.LT.0) GO TO 31
IF (ICPPOS.NE.0) GO TO 31
REWIND 3
IOFPCS = 1
IF (KFILE.GT.1) CALL SKPFL (KFILE-1,3)
C SET UP INPUT TAPE
31 CALL OCEAN1(NCHAN,INUNIT,MAXERR,INFILE,NTYPE,NSEARH)
      WRITE(6,35) IDUSER,NCHAN,KCHAN,NBLOCK,IBLOCK,SAMFRQ
      IF(IHANN.GT.0) WRITE(6,36)
      IF(LOFR.GT.0) CALL CONVOL(VAL,MXWAY,LDEC,1,SUM)
      DECIM = LDEC
      LINCR = IBLOCK
      IF(IBLOCK.LT.1024) LINCR = 1024
C SET UP STORAGE REFERENCES
      DO 40 K = 1,KCHAN
      KCH = K -1
40     INCR(K) = LINCR *KCH
      M,NCR = LINCR
      LOBLO = LOFR
      MBLOCK = 1
C READ SOME DATA
12     NPAR = 0
      IADD = 0
      N111 = 0
      7000 NOLD = 1
      SAMFRQ = DFREQ
      IF(LOBLO.GT.0) SAMFRQ = DFREQ/DECIM
      DO 289 IB = 1,IBLOCK
      48     IF(LOBLO) 49,49,101
      101    DO 150 LSAMP=1,LDEC
              CALL OCEAN2(ININD,KBLOCK,INDATA(1))
              NIND = NIND+1
              GO TO (140,70,116,105,140),NIND
      105    N111 = N111+1
              KCH = INDATA(1)/NCHAN
              ICH = INDATA(1)-(KCH*NCHAN)
              IF(!ICH) 51,106,51
      106    DO 108 IC = 1,KCH
              DO 107 K = 1,KCHAN
      107    VAL(NOLD,K) = 512.
              NOLD = NOLD+1
      108    CONTINUE
              GO TO 150
      116    IF (NZBLOK.EQ.KBLOCK) GO TO 7000
              NPAR = NPAR + 1
              NZBLOK = KBLOCK
              IF (INPAR.GT.MAXERR) GO TO 5000
              GO TO 7000

```

```

140 DO 145 K = 1,KCHAN
  IDSUB = MCHAN(K)
  IDSUB2 = MCHAN2(K)
  IF (NIND.NE.5) GO TO 7020
  DATA = ATA(IDSUB)*CAL(K) + ATA(IDSUB2)*CAL2(K)
  GO TO 7030
7020 DATA = (FLOAT(INDATA(IDSUB))/100. - 5.11)*CAL(K)
  DATA = DATA + (FLOAT(INDATA(IDSUB2))/100. - 5.11)*CAL2(K)
7030 VAL(NOLD,K) = DATA
145 CONTINUE
  NOLD = NOLD+1
  IF(NOLD.GT.MXWAY) NOLD = 1
150 CONTINUE
  IF(IADD.EQ.0.AND.NOLD.GT.1) GO TO 101
  DO 160 K = 1,KCHAN
    IDSUB = MCHAN(K)
    CALL CONVOL(VAL,NOLD,K,2,SUM)
    ATA(IDSUB) = SUM
160 CONTINUE
  NIND = 6
  GO TO 100
49 CALL OCEAN2(NIND,KBLOCK,INDATA(1))
C NIND = 1 MEANS END OF FILE, 2 MEANS PARITY ERROR, 3 MEANS ONES SEARCH
C NIND = 4 MEANS OCEAN2 WILL RETURN A FLOATING POINT NUMBER BETWEEN
C 0. AND 1023.
  NIND = NIND+1
  GO TO (100,70,60,50,100), NIND
50 N111 = N111+1
  KCH = INDATA(1)/NCHAN
  ICH = INDATA(1) - (KCH*NCHAN)
  IF ((ICH) 51,56,51
51 WRITE(6,55) MBLOCK,KBLOCK,IADD
  GO TO 5000
56 DO 58 IC = 1,KCH
  IADD = IADD+1
  DO 58 K = 1,KCHAN
    IWSUB = INCR(K)+IADD
    W(IWSUB) = 0.0
    GO TO 100
  60 IF (NZBLOK.EQ.KBLOCK) GO TO 7000
    NPAR = NPAR+1
    NZBLOK = KBLOCK
    IF (NPAR.GT.MAXERR) GO TO 5000
    GO TO 7000
70  WRITE(6,75)MBLOCK,KBLOCK
    GO TO 5000
C STORE DATA IN PSFUDD 2-D ARRAY
100 IADD = IADD+1

```

```

DO 200 K = 1,KCHAN
IDSUB = MCHAN(K)
IDSUB2 = MCHAN2(K)
IWSUB = INCR(K)+IADD
IF (NIND.LT.5) GO TO 7040
IF (NIND.EQ.5) DATA = ATA(IDSUB)*CAL(K) + ATA(IDSUB2)*CAL2(K)
IF (NIND.EQ.6) DATA = ATA(IDSUB)
GO TO 7050
7040 DATA = (FLOAT(INDATA(IDSUB))/100. - 5.11)*CAL(K)
DATA = DATA + (FLOAT(INDATA(IDSUB2))/100. - 5.11)*CAL2(K)
7050 W(IWSUB) = DATA
200 CONTINUE
289 CONTINUE
C NOW GET FOURIER TRANSFORM
299 MPOW = NPOW - 1
      CALL FORT(W1,MPOW,S,-2,IFERR)
      CALL USCRMB(W1,IBLOCK,S)
      IF (KCHAN.LE.1) GO TO 340
      KCH = MINCR/1024
      GO TO (320,300,300,301),KCH
300 KCH = KCHAN - 1
      GO TO (310,309,308,307),KCH
301 CALL FORT(W5,MPOW,S,-2,IFERR)
      CALL USCRMB(W5,IBLOCK,S)
      GO TO 340
320 KCH = KCHAN/2
      GO TO (306,305,304,303,302), KCH
302 CALL FORT(W10,MPOW,S,-2,IFERR)
      CALL USCRMB(W10,IBLOCK,S)
303 CALL FORT(W8,MPOW,S,-2,IFERR)
      CALL USCRMB(W8,IBLOCK,S)
304 CALL FORT(W6,MPOW,S,-2,IFERR)
      CALL USCRMB(W6,IBLOCK,S)
305 CALL FORT(W4,MPOW,S,-2,IFERR)
      CALL USCRMB(W4,IBLOCK,S)
306 CALL FORT(W2,MPOW,S,-2,IFERR)
      CALL USCRMB(W2,IBLOCK,S)
      KCH = (KCHAN+1)/2
      GO TO (340,310,309,308,307),KCH
330 KCH = (KCHAN-1)/2
      GO TO (310,309,308,307), KCH
307 CALL FORT(W9,MPOW,S,-2,IFERR)
      CALL USCRMB(W9,IBLOCK,S)
308 CALL FORT(W7,MPOW,S,-2,IFERR)
      CALL USCRMB(W7,IBLOCK,S)
309 CALL FORT(W5,MPOW,S,-2,IFERR)
      CALL USCRMB(W5,IBLOCK,S)

```

```

310 CALL FORT(W3,NPOW,S,-2,IFERR)
      CALL USCRM8(W3,IBLOCK,S)
340 IF (IFERR) 350,500,350
350 WRITE (6,355) IFERR
C COEFFICIENTS NOW UNSCRAMBLED, NOW BEGIN SUM AND PRINT
500 IF(IHANN.LE.0) GO TO 890
      DO 800 K = 1,KCHAN
      ISUB = INCR(K) + 1
      ISTAR = INCR(K) + 5
      ILSUB = INCR(K) + IBLOCK - 2
      KLR = ISUB + 2
      KLM = ISUB + 3
      AR3 = 0.0
      AR2 = W(ISUB)
      AR1 = W(KLR)
      AM3 = 0.0
      AM2 = 0.0
      AM1 = W(KLM)
      W(ISUB) = CORNA*(AR2-AR1)
      DO 700 KLP = ISTAR,ILSUB,2
      KM = KLP+1
      AR3 = AR2
      AR2 = AR1
      AR1 = W(KLP)
      AM3 = AM2
      AM2 = AM1
      AM1 = W(KM)
      W(KLR) = CORNB*((2.*AR2)-AR3-AR1)
      W(KLM) = CORNB*((2.*AM2)-AM3-AM1)
      KLR = KLP
      KLM = KM
700 CONTINUE
      ISUB = INCR(K)+2
      AR3 = AR2
      AR2 = AR1
      AR1 = W(ISUB)
      W(KLR) = CORNB*((2.*AR2)-AR3-AR1)
      W(ISUB) = CORNA*(AR1-AR2)
      AM3 = AM2
      AM2 = AM1
      AM1 = 0.
      W(KLM) = CORNB*((2.*AR2)-AR3-AR1)
CONTINUE
800 IF(IMPRINT.LT.0) GO TO 1300
      KCH = NPOW-6
      IF (KCH.LE.0) KCH = 1
      ISUM = I2TAB(KCH)
      KFIRST = 1

```

```

900      KLAST = 5
        IF(KCHAN.LT.KLAST)KLAST = KCHAN
        WRITE(6,905)IDUSER,MBLOCK,IBLOCK,TODAY,NPAR,N111
        IF(IHANN.GT.0) WRITE(6,36)
        DO 920 K = KFIRST,KLAST
        WRITE(6,915) K,MCHAN(K),(ACHNAM(L,K),L= 1,9),(AUNITS(L,K),L=1,2),CAL(K)
920      CONTINUE
        WRITE(6,925) (MCHAN(K),K=KFIRST,KLAST)
        WRITE(6,935)
        SU = 1
        IHAR = -1
        DO 1200 ICO = 1,33
        DO 950 K= KFIRST,KLAST
        KL = 2*K
        TAPRAY(KL)= 0.0
950      TAPRAY(KL+1) = 0.0
        DO 990 IS = 1,ISUM
        IHAR = IHAR+1
        IADD = 2*IHAR
        DO 980 K=KFIRST,KLAST
        KL = 2*K
        LINCR = INCR(K)
        ISTOR = LINCR+IADD
        IF(IHAR.EQ.(IBLOCK/2)) GO TO 970
        TAPRAY(KL) = TAPRAY(KL) + W(ISTOR +1)
        IF (IHAR.EQ.0) GO TO 980
        TAPRAY(KL+1) = TAPRAY(KL+1)+W(ISTOR+2)
        GO TO 980
970      TAPRAY(KL) = TAPRAY(KL)+W(LINCR+2)
980      CONTINUE
        IF (IHAR.EQ.0) GO TO 1000
990      CONTINUE
        SU = ISUM
1000      KONE = 2*KFIRST
        KTWO = (2*KLAST)+1
        DO 1010 KL = KONE,KTWO
1010      TAPRAY(KL) = TAPRAY(KL)/SU
        WRITE(6,1105) IHAR,(TAPRAY(KL),KL=KONE,KTWO)
1200      CONTINUE
        IF (KLAST.GE.KCHAN ) GO TO 1300
        KFIRST = KLAST + 1
        KLAST = KFIRST + 4
        GO TO 900
C END OF SUM AND PRINT LOOP, NOW WRITE OUT PUT TAPE
1300      IF(MTAPE:1350,1301,1301
1301      IF(LCBLO-1) 1302,1303,1302
1302      WRITE (3) ARTAPE
        GO TO 1304

```

```

1303 WRITE (3) ARTAFE
1304 CONTINUE
C SET UP 256 WORD TAPRAY
IHARM=(IBLOCK/2)-1
MAXTAP = 256-(2*KCHAN)
C ZEROTH HARMONIC IS A SPECIAL CASE
IHAR=1
IAR=0
DO 1310 K = 1,KCHAN
ISUB = INCR(K)
IHAR = IHAR+1
TAPRAY(IHAR) = W(ISUB+1)
IHAR=IHAR+1
1310 TAPRAY(IHAR) = 0.0
DO 1330 IH = 1,IHARM
IHH = 2*IH
IHAR = IHAR+1
TAPRAY(IHAR) = IH
DO 1320 K=1,KCHAN
ISUB = IHH + INCR(K)
TAPRAY(IHAR+1)=W(ISUB+1)
TAPRAY(IHAR+2)=W(ISUB+2)
IHAR = IHAR+2
1320 CONTINUE
IF(IHAR.LT.MAXTAP)GO TO 1330
IHAR=0
IF (LCBLO-1) 1322,1323,1322
1322 WRITE (3) TAPRAY
GO TO 1330
1323 WRITE (3) TAPRAY
1330 CONTINUE
C (IBLOCK/2)TH HARMONIC IS ALSO A SPECIAL CASE
IHAR = IHAR+1
TAPRAY(IHAR) = IHARM+1
DO 1340 K=1,KCHAN
ISUB = INCR(K)
TAPRAY(IHAR+1) = W(ISUB +2)
TAPRAY(IHAR+2) = 0.0
1340 IHAR = IHAR+2
IF(LCBLO-1) 1341,1342,1341
1341 WRITE (3) TAPRAY
WRITE(6,1355) MBLOCK,NFILE
GO TO 1343
1342 WRITE (3) TAPRAY
WRITE(6,13425) MBLOCK,IOUSER
1343 CONTINUE
1350 IF(LCBLC.NE.1) MBLOCK = MBLOCK+1
IF(MBLOCK.GT.NBLOCK) GO TO 5000

```

```

        IF(LOFR.EQ.0) GO TO 12
        IF(LOBLO-1)1351,1352,12
1351  LOBLO = 1
      GO TO 12
1352  LOBLC = 0
      GO TO 12
C WRAP UP ROUTINES
5000  IF(MTAPE.LT.0) GO TO 5500
      END FILE 3
      IF(LOFR.NE.1) GO TO 5500
      END FILE 2
      REWIND 2
      IDTEST = IDUSER
5200  READ (3,END=5300) ARTAPE
      WRITE (3) ARTAPE
      GO TO 5200
5300  END FILE 3
      NFILE = NFILE + 1
      IDUSER = IDTEST
      WRITE(6,5305) IDUSER,NFILE
5500  CONTINUE
      IF (MTAPE.GE.0) NFILE = NFILE + 1
      WRITE (6,8001)
      WRITE (6,8002)
      CALL OCEAN3
      GO TO 6000
6010  CALL RWUNL0
      IF (MTAPE.LT.0) GO TO 6020
      IF (IRWCTL.NE.0) GO TO 6030
      REWIND 3
6020  WRITE (6,5005)
      CALL EXIT
6030  REWIND 3
      GO TO 6020
6C70  WRITE(6,6075) MBLOCK,NFILE,LOBLO
6C75  FORMAT(5X,3CH END OF OUTPUT TAPE, MBLOCK = ,I5,9H NFILE = ,
1 I5,9H LOBLC = ,I5/5X,21HPROCESSING TERMINATED )
      GO TO 6010
5 FORMAT (I9,1X,5(I5,5X),F10.0)
6 FORMAT (7(I5,5X))
7 FORMAT (2I5,2F10.0,9A4,4X,2A4)
35  FORMAT(46H1 FOURIER TRANSFORM OF OCEAN TIME SERIES DATA /14H USE
1R NUMBER ,I11/16H OCEAN TAPE HAS,I5,1X,18HCHANNELS, OF WHICH ,
2 I5,16H WILL BE DONE IN ,I5,10H BLOCKS OF ,I6,13H SAMPLES EACH /
3 39H THE FREQUENCY OF DIGITAL SAMPLING WAS ,F10.2,9H SAMP/SFC/
36  FORMAT(64H0 FOURIER COEFFICIENTS TO BE HANNED AND NORMALIZED (*SQR
LT(8/3)) )
55  FORMAT(32H CHANNEL IDENTITY LOST IN BLOCK ,I5, 7H,RECORD,I5,

```

```

1 6H,SWEET,I7 )
75  FORMAT(36H END OF FILE ON INPUT TAPE IN BLOCK,I5,7H,RECORD,I5)
355 FORMAT(15H TOTAL IFERR = I10)
905 FORMAT(27H1 SUMMARY COEFFICIENTS FOR ,I10,1H,,I5,11HTH BLOCK OF,
1 I7,8H SAMPLES, 10X, /5X,37HIN THIS BLOCK PARITY ERRORS ON TAPE
2= ,I3,32H,ALL ONES NOT FOUND IN CHANNEL 1,I4,6H TIMES //16X,
3 1HK,4X, 7HCHANNEL,17X,4HNAME,25X,5HUNITS,3X,10HCAL.FAC.OR )
915 FORMAT(15X,I3,4X,I3, 5X,9A4,8X, 2A4,5X,1PE9.2)
925 FORMAT(2X,4HLAST,I1X,5(8X,8HCHANNEL ,I2,4X))
935 FORMAT(2X,4HHARM,3X,5(4HREAL,6X,4HIMAG,8X))
1105 FORMAT(2X,I4,1X,5(3X 'PE9.2,1X,1PE',2))
13425 FORMAT(24H COEFFICIENTS FOR BLOCK ,I5,36H OF SMOOTHED AND DECIMA
1 TED DATA FOR ,I9,14H WRITTEN ON 22 )
1355 FORMAT(24H COEFFICIENTS FOR BLOCK ,I5,17H WRITTEN IN FILE ,I3,
1 15H ON OUTPUT TAPE )
5005 FORMAT(13HNORMAL EXIT )
80  FORMAT(20(1X,F5.2))
5305 FORMAT(34H0 SMOOTHED AND DECIMATED DATA FOR ,I9,39H SUCCESSFULLY C
1 OPIED FROM 02 ONTO FILE ,I2,14H OF TAPE ON 03 /)
8001 FORMAT(42HINORMAL COMPLETION OF A PROCESSING REQUEST!
8002 FORMAT(1H )
FND

```

```

SUBROUTINE CONVOL(VAL,NULD,KCHAN,MODE,SUM)
DIMENSION VAL(50,1),WEIGHT(50)
IF(MODE-1) 10,10,20
10  LDEC = 10
MXWAY = 21
WAY = 1.0/21.0
DO 11 KM = 1,21
11  WEIGHT(KM) = WAY
KCHAN = LDEC
NOLD = MXWAY
WRITE(6,5) LDEC,MXWAY,(WEIGHT(K),K=1,MXWAY)
RETURN
20  SUM = 0.
ACCOUNT = 0
NFIR = NULD-1
30  NFIR = NFIR+1
ACCOUNT = ACCOUNT +1
IF(ACCOUNT-MXWAY)17,17,21
17  IF(NFIR-MXWAY)40,40,18
18  NFIR = 1
40  SUM = SUM+VAL(NFIR,KCHAN)
GO TO 30

```

```

21   SUM = SUM*WEIGHT(1)
      RETURN
5    FORMAT(50H1 ALTERNATE BLOCKS OF DATA DECIMATED BY FACTOR OF , I3/
1 2X,22H AFTER SMOOTHING WITH ,15,35H WEIGHTS WITH THE FOLLOWING VA
2LUES, /(10X,1PE14.5))
      END

SUBROUTINE USCRMB (C,M,S)
DIMENSION C(1),S(1)
N = M
ST = C(1)
C(1) = 0.5*(C(1) + C(2))
C(2) = 0.5*(ST - C(1))
K = N/2 - 1
MSIN = N/4
C(K+3) = -C(K+3)
DO 10 I=3,K,2
IS = (I-1)/2
JC = MSIN - IS
ST = S(IS)
CT = S(JC)
A1 = C(I)
B1 = C(I+1)
L = N - I
A2 = C(L+2)
B2 = C(L+3)
C(I) = 0.5*(A1+A2+(B1+B2)*CT-(A1-A2)*ST)
C(I+1) = 0.5*(B1-B2-(B1+B2)*ST-(A1-A2)*CT)
C(L+2) = 0.5*(A1+A2-(B1+B2)*CT+(A1-A2)*ST)
10 C(L+3) = 0.5*(B2-B1-(B1+B2)*ST-(A1-A2)*CT)
      RETURN
      END

```

FORTRAN OCEAN PACKAGE

THE FORTRAN OCEAN PACKAGE IS DESIGNED TO DUPLICATE THE 'OCEAN' AND 'OCEANB' SUBROUTINE PACKAGES WITH A TAPE WHICH WAS EITHER CREATED BY THE DIGITAL FILTERING SYSTEM OR IS IN THE SAME FORMAT. THE CALLS ARE EQUIVALENT.

THE CALLING SEQUENCES ARE:

CALL OCEAN1 (KCHAN,NUNIT,MAXERR,NFILE,NTYPE,NSEARH)

WHERE,

NCHAN = NO. OF CHANNELS DIGITIZED ON THE FORTRAN OCEAN TAPE

NUNIT = NO. OF UNIT ON WHICH THE FORTRAN OCEAN TAPE IS
MOUNTED

MAXERR (THIS PARAMETER IS NOT APPLICABLE TO FORTRAN OCEAN
BUT MUST BE INCLUDED FOR COMPATIBILITY)

NFILE = NO. OF DESIRED FILE ON THE FORTRAN OCEAN TAPE

NTYPE (NOT APPLICABLE BUT IS INCLUDED FOR COMPATIBILITY)

NSEARH (NOT APPLICABLE BUT IS INCLUDED FOR COMPATIBILITY)

CALL CCEAN2 (IND,KBLOCK,N)

WHERE,

IND = 0 FOR NORMAL RETURN (RETURNED)

= 1 FOR END OF INPUT FILE (RETURNED)

= 4 IF THE VECTOR RETURNED IS IN FLOATING POINT (FORTRAN
VERSION OF OCEAN ONLY) (RETURNED)

KBLOCK = THE SEQUENTIAL NUMBER OF THE PHYSICAL TAPE BLOCK OF
THE NEXT DATA WHICH WILL BE RETURNED BY THE NEXT OCEAN2 CALL.
(RETURNED)

N = FIRST LOCATION OF THE ARRAY INTO WHICH THE CROSS CHANNEL
SEQUENCE IS TO BE RETURNED.

CALL SECEANE

THIS CALL RESULTS IN A 1 LINE SUMMARY BEING LISTED ON LOGICAL
UNIT 6. THIS MAY BE CALLED AT ANY TIME.

```

SUBROUTINE OCEANI (NCHAN,NUNIT,MAXERR,NFILE,NTYPE,NSEARCH)
DATA KFILE/0/
KUNIT = NUNIT
IF (KFILE.GT.0) GO TO 20
10 REWIND KUNIT
KFILE = 1
KSTART = 1
20 IF (KFILE.GT.NFILE) GO TO 10
IF (KFILE.EQ.NFILE.AND.KSTART.EQ.0) GO TO 10
KSKIP = NFILE - KFILE
IF (KSKIP.LE.0) GO TO 50
DO 40 I=1,KSKIP
30 READ (NUNIT,END=40) IDUMMY
GO TO 30
40 CONTINUE
50 KFILE = NFILE
KSTART = 1
KMAX = 256
KCTR = KMAX + 1
KREAD = 0
RETURN
ENTRY OCEAN2 (IND,KBLOCK,X)
DIMENSION X(1),DATA(256)
IF (KCTR.GT.KMAX) GO TO 70
55 DO 60 I=1,NCHAN
X(I) = DATA(KCTR)
60 KCTR = KCTR + 1
IND = 4
KALOCK = KREAD
RETURN
70 READ (KUNIT,END=80) KMAX,NCHAN,(DATA(J),J=1,KMAX)
KSTART = 0
IF (KMAX.EQ.0) GO TO 70
KCTR = 1
KREAD = KREAD + 1
GO TO 55
80 KFILE = KFILE + 1
KSTART = 1
IND = 1
RETURN
ENTRY OCEAN3
WRITE (6,4001) KREAD
RETURN
4001 FORMAT ('OCEAN 3 CALLED (FORTRAN VERSION).',I10,' TAPE BLOCKS PRO
1CESSED')
ENTRY RWUNL0
REWIND KUNIT
KFILE = 1

```

```
KSTART = 1  
RETURN  
END
```

CCCCC THIS SUBROUTINE WHEN CALLED MOVES THE TAPE ON LOGICAL UNIT NUNIT PAST KSKIP END OF FILE MARKS. THE RECORDS SKIPPED OVER MUST BE IN FORTRAN BINARY. IF KSKIP IS ZERO OR NEGATIVE THE ROUTINE RETURNS WITHOUT MOVING THE TAPE.

```
SUBROUTINE SKPFL (KSKIP,NUNIT)  
IF (KSKIP.LT.1) RETURN  
DO 20 I=1,KSKIP  
10 READ (NUNIT,END=20) IDUMMY  
GO TO 10  
20 CONTINUE  
RETURN  
END
```

CCCCC FORT, ONE-DIMENSIONAL FINITE COMPLEX FOURIER TRANSFORM.

FORT 002

FORT 003

```
SUBROUTINE FCRT(A,M,S,IFS,IFERR)
```

FORT 004

FORT 005

FOURIER TRANSFORM SUBROUTINE, PROGRAMMED IN SYSTEM/360,
BASIC PROGRAMMING SUPPORT, FORTRAN IV. FORM C28-6504
THIS DECK SET UP FOR IBSYS UN IBM 7094.

FORT 006

FORT 007

FORT 008

THIS DECK MODIFIED TO ALLOW COMPUTATION OF SINE TABLE (S(J))
WITH M = 14, FOR USE WITH SERIFS OF 2**14 REAL VALUES
BY ADDITION OF STATEMENTS 6 AND 7 ANDCHANGING 3 FROM
IF(M-13) 5,5,2 TO IF(M-13) 5,5,5

DOES EITHER FOURIER SYNTHESIS, I.E., COMPUTES COMPLEX FOURIER SERIES
GIVEN A VECTOR OF N COMPLEX FOURIER AMPLITUDES, OR, GIVEN A VECTOR
OF COMPLEX DATA X DOES FOURIER ANALYSIS, COMPUTING AMPLITUDES.
A IS A COMPLEX VECTOR OF LENGTH N=2**M COMPLEX NOS. OR 2*N REAL
NUMBERS. A IS TO BE SET BY USER.
M IS AN INTEGER 0.LT.M.LE.13, SET BY USER.
S IS A VECTOR S(J)= SIN(2*PI*j/NP), J=1,2,....,NP/4-1,
COMPUTED BY PROGRAM.
IFS IS A PARAMETER TO BE SET BY USER AS FOLLOWS-
IFS=0 TO SET NP=2**M AND SET UP SINE TABLE.

FORT 009

FORT 010

FORT 011

FORT 012

FORT 013

FORT 014

FORT 015

FORT 016

FORT 017

FORT 018

FORT 019

IFS=1 TO SET N=NP=2**M, SET UP SIN TABLE, AND DO FOURIER
 SYNTHESIS, REPLACING THE VECTOR A BY
 X(J)= SUM OVER K=0,N-1 OF A(K)*EXP(2*PI*I/N)**(J*K),
 J=0,N-1, WHERE I=SQRT(-1)
 THE X'S ARE STORED WITH RE X(J) IN CELL 2*j+1
 AND IM X(J) IN CELL 2*j+2 FOR J=0,1,2,...,N-1.
 THE A'S ARE STORED IN THE SAME MANNER.
 IFS=-1 TO SET N=NP=2**M, SET UP SIN TABLE, AND DO FOURIER
 ANALYSIS, TAKING THE INPUT VECTOR A AS X AND
 REPLACING IT BY THE A SATISFYING THE ABOVE FOURIER SERIES.
 IFS=+2 TO DC FOURIER SYNTHESIS ONLY, WITH A PRE-COMPUTED S.
 IFS=-2 TO DC FOURIER ANALYSIS ONLY, WITH A PRE-COMPUTED S.
 IFERR IS SET BY PROGRAM TO-
 =0 IF NO ERROR DETECTED.
 =1 IF M IS OUT OF RANGE., OR, WHEN IFS=+2,-2, THE
 PRE-COMPUTED S TABLE IS NOT LARGE ENOUGH.
 =-1 WHEN IFS =+1,-1, MEANS ONE IS RECOMPUTING S TABLE
 UNNECESSARILY.
 NOTE- AS STATED ABOVE, THE MAXIMUM VALUE OF M FOR THIS PROGRAM
 ON THE IBM 7094 IS 13. FOR 360 MACHINES HAVING GREATER STORAGE
 CAPACITY, ONE MAY INCREASE THIS LIMIT BY REPLACING 13 IN
 STATEMENT 3 BELOW BY LOG2 N, WHERE N IS THE MAX. NO. OF
 COMPLEX NUMBERS ONE CAN STORE IN HIGH-SPEED CORE. ONE MUST
 ALSO ADD MORE DO STATEMENTS TO THE BINARY SORT KOUTINE
 FOLLOWING STATEMENT 24 AND CHANGE THE EQUIVALENCE STATEMENTS
 FOR THE K'S.
 DIMENSION A(1), S(1), K(14)
 EQUIVALENCE (K(13),K1),(K(12),K2),(K(11),K3),(K(10),K4)
 EQUIVALENCE (K(9),K5),(K(8),K6),(K(7),K7),(K(6),K8)
 EQUIVALENCE (K(5),K9),(K(4),K10),(K(3),K11),(K(2),K12)
 EQUIVALENCE (K(1),K13),(K(1),N2)
 IF(M)2,2,3
 3 IF(M-13) 5,5,6
 THE FOLLOWING TWO STATEMENTS ADDED BY J GARRETT , SEE ABOVE
 6 IF(M-14) 7,7,2
 7 IF(IFRS) 2,5,2
 2 IFERR=1
 1 RETURN
 5 IFERR=0
 N=2**M
 IF(IABS(IFRS) - 1) 200,200,10
 WE ARE DOING TRANSFORM ONLY. SEE IF PRE-COMPUTED
 S TABLE IS SUFFICIENTLY LARGE

FORT 020
 FORT 021
 FORT 022
 FORT 023
 FORT 024
 FORT 025
 FORT 026
 FORT 027
 FORT 028
 FORT 029
 FORT 030
 FORT 031
 FORT 032
 FORT 033
 FORT 034
 FORT 035
 FORT 036
 FORT 037
 FORT 038
 FORT 039
 FORT 040
 FORT 041
 FORT 042
 FORT 043
 FORT 044
 FORT 045
 FORT 046
 FORT 047
 FORT 048
 FORT 049
 FORT 050
 FORT 051
 FORT 052
 FORT 053
 FORT 054
 FORT 055
 FORT 056
 FORT 058
 FORT 059
 FORT 060
 FORT 061
 FORT 062
 FORT 063
 FORT 064

```

10 IF( N-NP )20,20,12 FORT 065
12 IFERR=1 FORT 066
C GO TO 200 FORT 067
SCRAMBLE A, BY SANDE'S METHOD FORT 068
20 K(1)=2*N FORT 069
DO 22 L=2,M FORT 070
22 K(L)=K(L-1)/2 FORT 071
DO 24 L=M,12 FORT 072
24 K(L+1)=2 FORT 073
C NOTE EQUIVALENCE OF KL AND K(14-L) FORT 074
C BINARY SORT- FORT 075
IJ=2 FORT 076
DO 30 J1=2,K1,2 FORT 077
DO 30 J2=J1,K2,K1 FORT 078
DO 30 J3=J2,K3,K2 FORT 079
DO 30 J4=J3,K4,K3 FORT 080
DO 30 J5=J4,K5,K4 FORT 081
DO 30 J6=J5,K6,K5 FORT 082
DO 30 J7=J6,K7,K6 FORT 083
DO 30 J8=J7,K8,K7 FORT 084
DO 30 J9=J8,K9,K8 FORT 085
DO 30 J10=J9,K10,K9 FORT 086
DO 30 J11=J10,K11,K10 FORT 087
DO 30 J12=J11,K12,K11 FORT 088
DO 30 J1=J12,K13,K12 FORT 089
IF( IJ-JI)28,30,30 FORT 090
28 T=A(IJ-1) FORT 091
A(IJ-1)=A(JI-1) FORT 092
A(JI-1)=T FORT 093
T=A(IJ) FORT 094
A(IJ)=A(JI) FORT 095
A(JI)=T FORT 096
30 IJ=IJ+2 FORT 097
IF(IFPS)32,2,36 FORT 098
C DOING FOURIER ANALYSIS, SO DIV. BY N AND CONJUGATE. FORT 099
32 FN = N FORT 100
DO 34 I=1,N FORT 101
A(2*I-1) = A(2*I-1)/FN FORT 102
34 A(2*I)=-A(2*I)/FN FORT 103
C SPECIAL CASE- L=1 FORT 104
36 DO 40 I=1,N,2 FORT 105
T = A(2*I-1) FORT 106
A(2*I-1) = T + A(2*I+1) FORT 107
A(2*I+1)=T-A(2*I+1) FORT 108
T=A(2*I) FORT 109
A(2*I) = T + A(2*I+2) FORT 110
40 A(2*I+2)=T - A(2*I+2) FORT 111
IF(M-1) 2,1 ,50 FORT 112

```

```

C      SET FOR L=2
50    LEXP1=2          FORT 113
      LEXP1=2** (L-1)   FORT 114
      LEXP=8            FORT 115
      LEXP=2** (L+1)   FORT 116
      NPL= 2**MT        FORT 117
      NPL = NP* 2**-L   FORT 118
60    DO 130 L=2,M     FORT 119
      SPECIAL CASE- J=0   FORT 120
      DO 80 I=2,N2,LEXP   FORT 121
      I1=I + LEXP1       FORT 122
      I2=I1+ LEXP1       FORT 123
      I3 =I2+LEXP1       FORT 124
      T=A(I-1)           FORT 125
      A(I-1) = T +A(I2-1)   FORT 126
      A(I2-1) = T-A(I2-1)   FORT 127
      T =A(I)             FORT 128
      A(I) = T+A(I2)       FORT 129
      A(I2) = T-A(I2)       FORT 130
      T = -A(I3)           FORT 131
      TI = A(I3-1)         FORT 132
      A(I3-1) = A(I1-1) - T   FORT 133
      A(I3 ) = A(I1 ) - TI   FORT 134
      A(I1-1) = A(I1-1) +T   FORT 135
      A(I1 ) = A(I1 ) +TI   FORT 136
      IF(L-2) 120,120,90    FORT 137
90    KLAST=N2-LEXP    FORT 138
      JJ=NPL              FORT 139
      DO 110 J=4,LEXP1,2   FORT 140
      NPJJ=AT-JJ           FORT 141
      UR=S(NPJJ)           FORT 142
      UI=S(JJ)              FORT 143
      ILAST=J+KLAST        FORT 144
      DO 100 I= J,ILAST,LEXP   FORT 145
      I1=I+LEXP1           FORT 146
      I2=I1+LEXP1           FORT 147
      I3=I2+LEXP1           FORT 148
      T=A(I2-1)*UR-A(I2)*UI   FORT 149
      TI=A(I2-1)*UI+A(I2)*UR   FORT 150
      A(I2-1)=A(I-1)-T       FORT 151
      A(I2 )=A(I ) - TI      FORT 152
      A(I-1) =A(I-1)+T       FORT 153
      A(I ) =A(I ) +TI       FORT 154
      T=-A(I3-1)*UI-A(I3)*UR   FORT 155
      TI=A(I3-1)*UR-A(I3)*UI   FORT 156
      A(I3-1)=A(I1-1)-T       FORT 157
      A(I3 )=A(I1 ) - TI      FORT 158
      A(I1-1)=A(I1-1)+T       FORT 159
      FORT 160

```

```

100 A(I1) =A(I1) +TI          FORT 161
END OF I LOCP               FORT 162
110 JJ=JJ+NPL                 FORT 163
END OF J LOCP               FORT 164
120 LEXP1=2*LEXP1             FORT 165
LEXP = 2*LEXP                FORT 166
130 NPL=NPL/2                 FORT 167
END OF L LOCP               FORT 168
140 IF(IF$)145,2,1           FORT 169
DO 150 I=1,N                  FORT 170
150 A(2*I) =-A(2*I)          FORT 171
160 GO TO 1                   FORT 172
RETURN                      FORT 173
MAKE TABLE OF S(J)=SIN(2*PI*j/NT),J=1,2,...,NT-1,NT=NP/4
200 NP=N                      FORT 174
MP=M                        FORT 175
NT=NP/4                     FORT 176
MT=4-2                      FORT 177
IF(MT) 260,260,205          FORT 178
205 THETA=.7853981634        FORT 179
THETA=PI/2***(L+1) FOR L=1
210 JSTEP = NT                FORT 180
JSTEP = 2***( MT-L+1 ) FOR L=1
JDIF = NT/2                  FORT 181
JDIF = 2***(MT-L) FOR L=1
S(JDIF) = SIN(THETA)         FORT 182
IF (MT-2)260,220,220         FORT 183
220 DO 250 L=2,MT            FORT 184
THETA = THETA/2.              FORT 185
JSTEP2 = JSTEP                FORT 186
JSTEP = JDIF                  FORT 187
JDIF = JDIF/2                 FORT 188
S(JDIF)=SIN(THETA)           FORT 189
JC1=NT-JDIF                  FORT 190
S(JC1)=COS(THETA)            FORT 191
JLAST=NT-JSTEP2              FORT 192
IF(JLAST-JSTEP)250,230,230   FORT 193
230 DO 240 J=JSTEP,JLAST,JSTEP
JC=NT-J                      FORT 194
JD=J+JDIF                    FORT 195
240 S(JD)=S(J)*S(JC)+S(JDIF)*S(JC)
250 CONTINUE                   FORT 196
260 IF(IF$)20,1,20            FORT 197
END                          FORT 198
                                FORT 199
                                FORT 200
                                FORT 201
                                FORT 202
                                FORT 203
                                FORT 204
                                FORT 205

```

C SPECTRUM AND CROSS SPECTRUM STATISTICS FROM FOURIER
COEFFICIENT TAPE PRODUCED BY FT4CR PROGRAM ON IBM
SYSTEM 360 MODEL 67

LAST REVISION JANUARY 27, 1969

JOHN GARRETT

THIS PROGRAM READS FOURIER COEFFICIENTS FROM TAPE PRODUCED BY FT4CR
PROGRAM, AND FROM THE APPROPRIATE SUMS OF THESE PRODUCES SPECTRA AND
COSPECTRA. THE SUMS USED MAY BE FIXED WITH FREQUENCY OR MAY GO IN HALF
OCTAVES. FROM A SPECIFIC LOW FREQUENCY OR AN AVERAGE, STANDARD DEVIATION
AND LINEAR COEFFICIENT ARE GIVEN FOR EACH BLOCK USED (SEE
FT4CR DESCRIPTION). THE COEFFICIENTS ARE GIVEN FOR EACH
AT EACH FREQUENCY. THE COSPECTRUM VALUE OF SPECTRAL DENSITY. (1) AND (2) IS GIVEN BY
AND THE SPECTRUM BY
$$\text{WHERE } \{R(N)\} = \{I(N)\}^2 + \{Q(N)\}^2$$

$$\text{AND } \{R(2)\} = \{I(2)\}^2 + \{Q(2)\}^2$$

$$\text{AND } \{R(1)\} = \{I(1)\}^2 + \{Q(1)\}^2$$

$$\text{AND } \{R(-1)\} = \{I(-1)\}^2 + \{Q(-1)\}^2$$

$$\text{IS THE COMPLEX FOURIER COEFFICIENT}$$

A VARIETY OF PLOTTED OUTPUT IS AVAILABLE. IN ALL A HORIZONTAL BAR
INDICATES THE FREQUENCY INTERVAL INCLUDED IN THE ESTIMATE PLOTTED. AND
A VERTICAL BAR INDICATES THE EXPECTED STANDARD DEVIATION OF THE
ESTIMATE (= STD. DEV. OF BLOCKS AVERAGED TO GIVE ESTIMATE/ SQRT(NUMBER
OF BLOCKS))

THE FOLLOWING SUBROUTINES MUST BE SUPPLIED BY USER
PHASES, TIC, LABEL
PLVAL, TIC, LABEL

THE FOLLOWING LOGICAL INPUT/OUTPUT UNITS ARE USED BY THIS PROGRAM
3= (TAPE) SUPPLYING COEFFICIENTS AND IDENTIFICATION AS
4= PRCCDUS BY FT4CR
5= (CARDS) CONTROL PARAMETERS
6= PRINTED OUTPUT

THE FOLLOWING INPUT IS REQUIRED

A CARD IS REQUIRED TO IDENTIFY YOUR GRAPHICAL OUTPUT FOR THE COMPUTING
CENTER STAFF. IT MUST BE PRESENT WHETHER PLOTS ARE PRODUCED OR NOT.
THE FIRST 72 COLUMNS OF THIS CARD WILL BE REPRODUCED ON THE BEGINNING
OF YOUR PLOT. THIS CARD ONLY APPEARS IN THE JOB AND IS THE FIRST
DATA CARD. THE FOLLOWING SET OF CARDS IS PRESENT FOR EACH FILE OF
FOURIER COEFFICIENTS TO BE PROCESSED.

FIRST CARD IN COLUMNS
1-9 ICUSER = USER IDENTIFICATION NUMBER FOR DATA DESIRED
14-15 ICMAX = NUMBER OF CHANNELS TO BE USED (MAX 10)

24-25 IBMAX = NUMBER OF DATA BLOCKS FOR STATISTICS
 34-35 19STAR = DATA IS IN NFILE-TH FILE ON FOURIER COEFFICIENT TAPE
 44-45 NFILES = DATA IS IN (BLANK) IF PLOTE AXES TO BE SET BY PROGRAM
 54 IAIXIS = 0 IF AXES TO BE SET BY USER
 55 IPLOT = 1 IF NCPLT OUTPUT DESIRE
 = 1 IF SPECTRAL DENSITY TO BE PLOTTED AGAINST FREQUENCY
 = 2 IF LCG10 SPECTRAL DENSITY TO BE PLOTTED AGAINST LOG10
 = 3 IF LCG10 SPECTRAL DENSITY TO BE PLOTTED AGAINST LOG10
 = 4 IF (FREQ* SPECTRAL DENSITY) TO BE PLOTTED
 = 5 SAME AS IPLOT=1 EXCEPT COHERENCE AND PHASE PLOTTED
 = 6 SAME AS 2 EXCEPT QUAD SPECTRA
 = 7 SAME AS 3 EXCEPT COHERENCE AND PHASE PLOTTED INSTEAD
 OF CO+QU
 = 8 SAME AS 4 WITH COHERENCE AND PHASE INSTEAD OF CO+QU
 65 IPHASE = 0 FOR PHASE CORRECTIONS
 = 1 FOR PHASES TO BE CORRECTED.

CAFE INSERTED ONLY IF IAIXIS = 1
 1-10 VALUE OF ORIGIN FCP SPECTRUM AND COSPECTRUM AXIS (F10.)
 11-20 UNITS PER INCH FOR SPECTRUM AND COSPECTRUM AXIS (F10.)
 21-30 VALUE OF ORIGIN FOR QUAD SPECTRUM AXIS LONG
 31-40 UNITS PER INCH FOR QUAD SPECTRUM AXIS (F10.) AXIS WILL BE
 SCALING INCHES LONG. (COHERENCE AND PHASE WILL BE
 SCALING BY PROGRAM. IF PLOTTED INSTEAD OF CU AND QUAD.)
 41-50 VALUE OF ORIGIN FOR FREQUENCY AXIS (F10.)
 51-60 UNITS PER LENGTH OF FREQUENCY AXIS IN INCHES (F10.)
 NOTE WHEN LCG10 PLOTS HAVE BEEN REQUESTED NUMBERS ABOVE MUST ALL
 REFER TO LCG10. E.G. UNITS PER INCH = 1.00 MEANS 1 DECADE PER INCH

NEXT CARD, IN COLUMN LINCOLN = 0 FOR CONSTANT BANDWIDTH GIVEN BY BANDW BELLOW
 LINCOLN = 1 FOR EXPONENTIAL BANDWIDTHS (GIVES LOGIC FREQ.
 6-15 STFRFC = APPROXIMATE CENTER FREQUENCY OF FIRST POINT OF
 SPECTRUM. IN CENTERZ (MUST INCLUDE ZERO TH HARMONIC IF STFRFC
 IS LESS THAN BANDW) WILL INCLUDE ZERO TH HARMONIC IF STFRFC
 15-25 BANDW = APPROXIMATE BANDWIDTH FOR FIXED BANDWIDTH SPECTRA,
 3C INCDK = 1 IF FOURIER COEFFICIENTS TO BE HANNEC BEFORE
 SPECTRA COMPUTED

4-5 SUBSEQUENT ICNAM NUMBER OF A CHANNEL FOR WHICH SPECTRA ARE WANTED
 CHANNEL IS USUALLY 25-29-30. CHANNEL NUMBERS FOR CONVERSION
 14-15, 15-20, 24-25 FOR WHICH CROSS SPECTRA WITH CHANNELS GIVEN IN CROSS
 CHANNELS ARE DESIRED. CHANNELS WITH CHANNELS GIVEN IN CROSS
 SPECTRUM ARE CROSS SPECTRUM. IF THE NUMBER IN 45
 SPECTRUM OF CHANNEL IS ITSELF. IF THE NUMBER IN 45
 SPECTRUM OF CHANNEL IS ITSELF. IF THE NUMBER IN 45
 SPECTRUM OF CHANNEL IS ITSELF. IF EACH CHANNEL APPEARS IN THE LIST
 SPECTRA OTHER CHANNELS SCARD GET SPECTRUM FOR
 CHANNEL 2 AND CROSS SPECTRUM BETWEEN 2 AND CHANNEL 8 THERE
 MUST BE A CARD WITH 2 IN COL. 5 AND BOTH 2 AND 8 IN THE LIST
 AND ANOTHER CARD WITH 8 IN COL. 5 AND BOTH 2 IN THE LIST.

PHASE CORRECTION DECK(INSERT ONLY IF PHASE NOT ZERO)
 PHASE CORRECTIONS APPLIED AT FREQUENCIES LISTED BELOW
 FIRST CARD IS LESS THAN 6
 1-10, 11-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80, 81-90, 91-100 FREQUENCIES (F1C-4) AT WHICH PHASE
 CORRECTIONS ARE TO BE SUPPLIED (UP TO 48 OF THEM)
 IF THE LAST FREQUENCY IS LESS THAN THE FULL 48 FREQUENCIES ARE SUPPLIED. THEN

SUBSEQUENT ICNAM SETS OF K PHASE CORRECTIONS (F10-4) TO BE APPLIED
 AT THE ABOVE FREQUENCIES. EACH SET CONTAINS THE CORRECTIONS AS THE
 FIRST CARD AND THE SETS ARE TO BE APPLIED TO A
 SPECTRUM CARDS. IF THE CORRECTIONS ARE TO BE APPLIED TO A
 CHANNEL THE K CARDS FOR THAT CHANNEL SHOULD BE BLANK. A PCSITIVE CORRECTICA WILL
 CAUSE THE CORRECTED PHASE TO LEAD THE UNCORRECTED ONE.

LAST CARD MUST BE BLANK UNLESS ANOTHER FILE IS TO BE PROCESSED, IN
 WHICH CASE A COMPLETE NEW SEQUENCE OF CARDS APPROPRIATE TO THE NEW
 FILE SHOULD FOLLOW.

DIMENSION ARTAPE(256), TAPRAY(256), PHI(10,10,32), PHISQ(10,10,32),
 1, BAND(10,10,32), A(10,10,32), KCH(10,10,32), B(10,10,32), END(33),
 2, CAL(10,10,32), MCHAN(10,10,32), INDCH(10,10,32), D(10,10,32),
 3, POS(10,10,32), DC(10,10,32), IDCSQ(10,10,32), CCBL(10,10,32),
 4, PC(10,10,32), AR(10,10,32), AM(10,10,32), TAB(10,10,32),
 5, ARI(10,10,32), AR2(10,10,32), BR(10,10,32), AM2(10,10,32),
 6, BR2(10,10,32), BR3(10,10,32), AM3(10,10,32), BR1(10,10,32),
 7, COSP(10,10,32), CUSP(10,10,32), POSL(10,10,32), BM3(10,10,32),
 8, CUSP(10,10,32), POSL(10,10,32), PCSU(32), NLGFW(16),
 9, EQUIVALENCE(TAPRAY(100), ICH(11), TAPE(11), Z(11), IDUSER),
 10, Z(2), NBLOCK, (Z(3), NTAP), (Z(5), KCHAN), (Z(11), MCHAN(11)),
 11, (Z(6), SANFRG), (Z(11), MCHAN(11)), (Z(31), ACHAN(11,1))


```

KB = C
KB = 39 J = I1C
IF (KCHA(KA).GT.ICH(KA,J)) GC TC 39
KB = KB+1
KCHA(KA,KB) = ICH(KA,J)
CONTINUE
39
40
CONTINUE
CALL IPHASE(EC,O,GC,TO,41
KICHMAX,IPCHAN,MAXFILE,QU,IFB,FREQ,NEND,BANDW,PHASE,KCA,KCB,FUNDFR,
1
41 KSKIF = INFILE - IFILE
IF (KSKIF.GT.0) CALL SKPFL (KSKIP,3)
IFILE = INFILE
IF (3,ENC=1108) ARTAPE
READ(3,ENC=1108) ARTAPE
KBEGIN = C
KIF(IFEST,-1)USER) 4900,54,4900
42 KIF(IFEST,AP - NALCCK) 59,65,55
54 55 CUC56 N = 1*AP
56 READ(3,END=1108) TAPRAY
GC TC 54
57 GC TC = (IBSTAR + IBMAX-1)-MBLOCK
58 IF ((IBSTAR + IBMAX-1)-MBLOCK
59 IBSTAR = 47CC,6C16
60 IBMAX = IBMAX - MBLOCK
61 IBSTAR = MBLOCK
62 IBSTAR = MBLOCK
C PRINTS = C
63 WRITE(6,62) IDUSER,IFILE,ICMAX,IMAX,IBSTAR
IF (IFILE.NE.0) INDOW = 0
C EXCHANGE CHANNEL NUMBERS FOR INDICES USED ON TAPE
64 COT = 67 K = 1,10
65 COT = C
KC = KCHA(K)
KIS = K - IS
KIF((KC*LE*0) GC, TO, 655
KCHA(KIS) = INDCH(KC)
IF ((INDCH(KC)*EQ.0) IS = IS+1
66 COT = 66 I = 1,10
67 COT = KCHE(KI)
KIT = I - IT
KCHB(KIS,KIT) = 0
IF ((KC*LE*0) GC, TO, 56
KCHB((KIS,KIT) = INDCH((IC))
IF ((INDCF(IC)).EQ.0) IT = IT+1
68 CONTINUE
69 CCHAR = 256 / ((2*KCHAN)+1)

```

```

C SET UP SUMMING FACULTY. FIRST GETTING FUNDAMENTAL FREQUENCY
JBLOCK = TBLOCK/2
TBLOCK = TBLOCK
FUNDFR = SAWFRQ/TBLOCK
TBLKLAG = TBLOCK/SAWFR
STRFRQ = STRFRQ/FUNCFR
TFRSCL=C
TFCLINECT, TFCNICS FOR FIXED BANDWIDTH
NUMBER = BAND/FUNCFR
IFINBAND=E.C) NBAND=1
BAND = NBAND*FUNCFR
BANDAF = BAND/2.0
J = NBAND/2
JSTART = JSTART-NSTART
JEND = NSTART+NBAND-1
I SET NUMBER CF FIXED BANDS
MAXIF = (JBLOCK-NSTART)/NBAND
ISTRFRG = NSTART
ISTRFRG = NSTART+NBAND
FREQ(1) = STRFRG*FUNCFR
FOSC(1)=FREC(1)
FOSC(1)=FREC(1)
FOSC(1)=FREC(1)+NBAND-1
BAND(1) = BAND(1)-BANHAF
FCSU(1) = FREC(1)-BANHAF
FCSU(1) = FREC(1)+BANHAF
FCSU(1) = 2, MAXIF
FREQ(1)=FREC(J) + BAND
FOSC(1)=FREC(1)
FOSC(1)=FREC(1)+BANHAF
FOSC(1)=PCSU(J)
FOSC(1)=BAND(J)+NBAND
NBAND=C80
SETUP FOR LOGARITHMIC SUMMING
IEND(1)=1
IEND(1)=FLACFR/2.0
IEND(1)=FLACFR+POS(1)
IEND(1)=SCRT(PCSU(1)*PCSU(1))
FOSC(1)=FREC(1)
FOSC(1)=FREC(1)
CEND(1)=ALGBW(1)
BAND = NBAND

```

73
74

```

FAND = RAND+C.50
I = S = I+1
BAND = BANC*I.33352
END(I) = BANC
MAXIF = I
IF(NEND(I)*LT(JBLOCK)) GC TC 77
IEND(I) = JBLOCK
EC 78 I = 2,MAXIF
J = I-1
EAND = NEND(I)-NEND(J)
FAND = BAND*FUNDFR
BAND = NEND(I)
PUSU(I) = (BAND+C.50)*FUNDFR
PUSU(J) = SQRT(FCSL(I)*FCSU(I))
PUSU(ISET) = FREQ(I)
78 EC NON XBLOCK = IF(AN*GT*C*AND*NEND(MAXIF)*EC,JBLOCK) NEND(MAXIF) = JBLOCK - 1
EC 61 IF(STARK = NSTART-INDW
IFR=1,MAXIF
C100 KA = 1.10
KA*KB = 1.10
C101 KQ(KA,KB) = C.0
C102 KB(KA,KB) = C.0
C103 BLK(KA,KB,IFR) = 0.0
C104 HSQ((KA,KB,IFR)) = C.0
C105 HIBL((KA,KB,IFR)) = C.0
C106 COUNTINUE
C107 CBL = C.0
C108 ELSQ = Q.0
C109 ELOCK = NBLCK
C110 DCL181 XBC = 1,10
AR11(KBC) = C.0
AR21(KBC) = C.0
AR31(KBC) = C.0
AR41(KBC) = C.0
AR51(KBC) = C.0
AR61(KBC) = C.0
AR71(KBC) = C.0
AR81(KAC,KBC) = 1,10
AR91(KAC,KBC) = 0.0
AR101(KAC,KBC) = 0.0
AR111(KAC,KBC) = 0.0
AR121(KAC,KBC) = 0.0
C111 CBEGIN = Q.0
C112 ELOCK = NBLCK
C113 LCP
C114 DCL181 XBC = 1,10
AR11(KBC) = C.0
AR21(KBC) = C.0
AR31(KBC) = C.0
AR41(KBC) = C.0
AR51(KBC) = C.0
AR61(KBC) = C.0
AR71(KBC) = C.0
AR81(KAC,KBC) = 1,10
AR91(KAC,KBC) = 0.0
AR101(KAC,KBC) = 0.0
AR111(KAC,KBC) = 0.0
AR121(KAC,KBC) = 0.0

```



```

196      AR3(KA) = AR2(KA)
          AR2(KA) = TAPRAY(IIS)
          AR1(KA) = AR2(KA)
          AR3(KA) = AR1(KA)
          AR2(KA) = TAPRAY(IIS+1)
          IFC(IFAR#2*LT*START) GC TC 197
          IFL = CCRNB*((2*0*AR2(KA))-ARI(KA)-AR3(KA))
          ANG TC 195
          ARL = CCRNB*((2*(C*AN2(KA))-ANI(KA)-AN3(KA)))
          CCNT = 0*C
          ARL = 0*C
          CCNT = IUE
          CCU 97C KB = 1*ICMAX
          KCB = KCBE(KA,KB)
          IFL(KCB*LE*2) GC TC 97C
          IT = INC(J)*CT*J) GC TC 206
          ERL = TAPRAY(IT)
          ERG = TAPRAY(IT+1)
          GC TC 205
          BR3(KA,KB) = PR2(KA,KB)
          PR2(KA,KB) = PR1(KA,KB)
          BR1(KA,KB) = TAPRAY(IT)
          BR3(KA,KB) = PR2(KA,KB)
          ER2(KA,KB) = PR1(KA,KB)
          ER3(KA,KB) = PR2(KA,KB)
          ER4(KA,KB) = TAPRAY(IT+1)
          IFL(IFAR#2*EC*START) GC TC 207
          IFL = CCRNB*((2*0*BR2(KA,KB))-BR1(KA,KB)-BR3(KA,KB))
          ERG = CCRNB*((2*0*BR2(KA,KB))-BR1(KA,KB)-BR3(KA,KB))
          GC TC 209
          ERL = 2*C*CCRN#*(PR2(KA,KB)-ER1(KA,KB))
          ERG = 0*C
          CCNT = IUE
          CC = ((ARL*ANG)+(ANG*BNG))/2.0
          CU = ((ARL*ANG)-(ARL*BNG))/2.0
          C STORE REAL PARTS
          IFC(IFAR#2*NEC) GC TC 901
          IFC(KCA*EC*KCB) GC TC 250
          DCSQ(KCA*KCB) = ARL*BRL*DC(KCA,KCB)
          DCSQ(KCA*KCB) = 4*D0*CC*CO+DCSC(KCA,KCB)
          DCSQ(KCA*KCB) = ARL*ERL*BLCK+DCBL(KCA,KCB)
          DCSQ(KCA*KCB) = ARL*DC((KCA,KCB))
          DCPL(KCA*KCB) = ARL*ARL*DCS((KCA,KCB))
          DCPL(KCA*KCB) = (ARL*BLCK)+DCBL(KCA,KCB)

```


SUCSS FORMAT (374-C RUN TERMINATED IN AN OGRELY MANNER)
CONTINUE
END

C SUBROUTINE FLYAL(A,AD,AE,PCS,FCSL,PCSL,NVAL)
C SPLCT =FLYAL AND LABEL
C AUGUST 61966
C LINES WHICH A(1)=AE(1),AC(1),PCS(1),FCSU(1),POSL(1)
CC 20 K = 1 NVAL
Y = A(K) + 3 * 5C
X = PCS(K) + 2 * 0C
XF = PCSU(K) + 2 * 0C
CALL = FLCUT(XF,Y,+3)
XP = PCSU(K) + 2 * 0C
CALL = FLCUT(XF,Y,+2)
Y = AE(K) + 3 * 5C
CALL = FLCUT(X,Y)
Y = AE(K) + 3 * 5C
CALL = FLCUT(X,Y,+2)
CALL TIC(X,Y)
CONTINUE
RETURN
END

20

51

SUBROUTINE TIC(X,Y)
SX = X+0 * C2
CALL FLCUT(X,Y,+2)
X = X-C * C4
CALL FLCUT(X,Y,+2)
X = X+0 * C2
CALL FLCUT(X,Y,+2)
RETURN
END

C VERSION CF APRIL 26 1968

```
C SUBROUTINE LABEL (U1,U2,ISY)
C
C DIMENSION U1(2),U2(2)
C
C X = 1.27C
C Y = 7.3C
C T = C/14
C CALL SYMBCL(X,Y,HT,1H(.95.0,1))
C Y = Y+0.12
C CALL SYMBCL(X,Y,HT,UL.90.0,8)
C Y = Y+1.00
C GCTC (10.2C) ISY
C CALL SYMBCL(X,Y,HT,1HX,9J.0,1)
C Y = Y+0.20
C CALL SYMBCL(X,Y,HT,UL.90.0,8)
C Y = Y+C.96
C GCTC 30
C XSUP = X-SUP
C CALL SYMBCL(XSUP,Y,HT,1H2,90.0,1)
C Y = Y+0.18
C CALL SYMBCL(X,Y,HT,7H/HERTZ),90.0,7)
C RETURN
C END
```

```
C
C SUBROUTINE PHASESIN,ACC,QU,IFB,F,NC,BW,PHASE,KCA,KCB,FFR,ICM.
C
C 1 MCHAN, MAXIF, RAD, KCHA, PCHAN(1), PCHAN(11), PHASE(50,11), FRED DCBSON,
C 1 SUMF(12), MNR(10), KCHA(11), PCHAN(11), PHASE(50,11), PHCOR(5C,11),
C 1 KTRL = N + 1
C GCTC (10.1C,500) , KTRL
C GCGC 98C0 NZERC = 1.48
C GCGC 98C0 NZERC = 1.11
C FCUR(1)NZERC, NZERO ) = 0.0
C CONTINUE
C FRE = 0
C GCGC 9820 KARC = 1.6
C READ(5,9825) (PHASE(JFRE,1), JFRE = 1,8 )
C GCGC 9820 LFR = 1.8
C FFR = MFR + 1
C FIN = PHASE( LFR,1 ) GCE,1 )
C FFCOR( FFR,1 ), FIN = FIN
C CNT(1) = 48
C NFR = 48
```

983C
 GC TC SS25
 KFK = NFK - 1
 NCKH = IGN + 1
 NCKX = 3
 NCKY = 40
 KKC = 1
 KAR = 1
 KASE (JFRE, NCHS), JFRE = 1,2)
 NCCR = 1
 LCCR = 1
 NCCR = NCCR + 1
 NCCR, NCFS) = PHASE(LCCR, NCHS)

983E
 GC TC SS25
 KFK = NFK - 1
 NCKH = IGN + 1
 NCKX = 2, NCH
 NCKY = 40
 KKC = 1
 KAR = 1
 KASE (JFRE, NCHS), JFRE = 1,2)

984U
 REPH = 1
 KPH = 45C
 LS = 1, MAXIF
 CUSUMPLS) = C, 1

110
 CANTIAUE = 9W(1) / FFR
 ANIB = ANIB + C * 30.0001
 KIB = KIB * EC
 KIB = KIB * 1
 KIB = 1
 FASE(1) = F(1)
 F = KAC(1) - KIB
 CO = 40C
 CO = 7 = 1, KIB

C = E + F
 FRS = C * FCF2
 IFC(FRS * GT * FK) GC TC 400
 AFL = AFL + 1
 AFL = AFL * GT * FR) GS TC 400
 FRL = AFL * GT * FR) AFL GC TC 13C
 GUPH TC 140 + 1

120
 C = E + F
 FRS = C * FCF2
 IFC(FRS * GT * FK) GC TC 400
 AFL = AFL + 1
 AFL = AFL * GT * FR) GS TC 400
 FRL = AFL * GT * FR) AFL GC TC 13C
 GUPH TC 140 + 1

130
 C = E + F
 FRD = FRL - FRS
 FFRFR = FC1F / FRD
 CUCIJSO = 21PCH
 FHJSD = PFLC(R(KPH))
 FHJSD(1) = SUPP(H) + PHIS + PHIC*FRFR

140
 C = E + F
 FRD = FRL - FRS
 FFRFR = FC1F / FRD
 CUCIJSO = 21PCH
 FHJSD = PFLC(R(KPH))
 FHJSD(1) = SUPP(H) + PHIS + PHIC*FRFR

390
 CUCIJSO = 45EC
 CUCIJSO = 2, NCH
 CUCIJSO = SUMP(MP) / (ANIB)

400
 CUCIJSO = 45EC
 CUCIJSO = 2, NCH
 CUCIJSO = SUMP(MP) / (ANIB)

450
 CUCIJSO = 45EC
 CUCIJSO = 2, NCH
 CUCIJSO = SUMP(MP) / (ANIB)

THIS SUBROUTINE WHEN CALLED MOVES THE TAPE ON LOGICAL UNIT NUNIT
 FAST KSKIP ENCL FILE MARKS. THE RECORDS SKIPPED OVER MUST BE IN
 FORTRAN BINARY. IF KSKIP IS ZERO OR TAPE IS ZERO THE ROUTINE RETURNS
 WITHOUT MOVING THE TAPE.

```

SUBROUTINE SKPFL (KSKIP, NUNIT)
IF (KSKIP .LT. 1) RETURN
1C READ (NUNIT, END=20) ICURRY
2C CCRTRC10
3C RETURN
END
  
```

```

SUBROUTINE SCALE (X, A, S, YMIN, DY, K)
DIMENSION X(2)
LOGICAL FLAG.
YMAX=X(1)
YMIN=YMAX
NP=N#K
DO (YMAX-X(1)) 5,6,6
5 IF (X(1)-YMIN) 7,10,10
7 IF (X(1)-X(1))
10 RANGE=YMAX-YMIN
ICY=C0
DO 12 I = 1, NP, K
12 X(I)=C0
13 RETURN
11 SFACUT = RANGE / (SFAC)
11 SFACCC = ALG10(SFACT)
11 SFACFLG = GE000) GU TC 13
11 SFACCC = SFACCG - 1.
13 SFACUT = SFACCT / 10.* SFAC FLG
  
```

```

14 IF(SFACT .GT. 1.5) GO TO 14
15 CY = TC * ISFLCG
16 CCU TC * ISFLCG
17 CY = 2.0 * ISFLCG
18 IF(SFACT .GT. 4.0) GO TO 17
19 CY = 4.0 * ISFLCG
20 CY = 1.5 * ISFLCG
21 CY = 1.0 * ISFLCG
22 CY = 1.0 * ISFLCG + 1
23 TEMP = YMIN / CY
24 IF(TEMP .LT. 0) TEMP = TEMP - 1.0
25 IYMIN = FLCTE(1YMIN) * CY
26 FLAG = FLCTLE
27 TEMP = YMIN + S * DY
28 SCAL = YMAX - YMIN
29 IF(YMAX .GT. TEMP) TEMP = TEMP - 1.0
30 X(I) = (X(I) - YMIN) / DY
31 RETURN

```

R. L. BROWN

```

SUBROUTINE AXIS(X,Y,PCD,NC,SIZE,THET,A,YMIN,CY)
C
SIGNSIGN BCD(2,1)
SIG = 1.0
SIG = -1.0
1 NC = 1425( NC )
2 TH = 1425( NC )
3 SIZE + C.5 * C.17 + 53264
4 N = SIZE + 1
5 TN = N
6 CTH = CCS( TH )
7 STH = SIN( TH )
8 SCALE THE AXIS *****
9 CY = CY
10 ABSU = YMIN
11 EXP = 0.0
C***SCALE THE AXIS *****

```

```

C
12    IF( ACY .GT. 9 ) ACY = 10*C + 10*C + 12 , 12
      ACY = ACY * C+1
      ABSU = ABSL * C+1
      EXP = EXP + 1.0
      GC_TC = 0
14    ACY = ACY * 10*C
      ABSU = ABSL * 10*C
      EXP = EXP - 10*C
      IF( ABS( ACY ) - 0.01 ) LOC_P** = 16 , 16
C*** ** INITIALIZE THE MAIN LOC_P** FOR THE LINE
C     XL , YL ETC. ARE FOR THE NUMBERS
C     XN , YN ETC. ARE FOR THE TITLE
C*** **
18    XS = X
      TEMP = XS - 0.2*SIG - J*C5
      XNA = XS - TEMP*STH - 0.0857*CTH
      NAFT = 3*ABS( ABSU ) * 1.000*C + C*5
      ICK = NOD( ICK , 100 ) * EC*C ) NAFT = 2
      IF( NOD( ICK , 100 ) * EC*C ) NAFT = 1
      CALL NUMBER( XNA , YNA , C+1C , ABSU , THETA , NAFT )
      XLB = XS + TN * CTH
      YLB = XS + TN * STH
      TEMP = 0.1*SIG
      TXLA = XLP - TEMP * STH
      ABY2 = YLP + TEMP * CTH
      C*** ** THIS IS THE MAIN LCCF ****
      C*** ** DC 20 I = NUMBER ****
      C*** ** NEXT THE NUMBER ****
      XNA = XNA + CTH
      YNA = YNA + STH
      ABSUT = ABSU + ACY
      NAFT = 3*ABS( ABSU ) * 1.000*C + C*5
      ICK = NOD( ICK , 100 ) * EC*C ) NAFT = 2
      IF( NOD( ICK , 100 ) * EC*C ) NAFT = 1
      CALL NUMBER( XNA , YNA , 0.10 , ABSU , THETA , NAFT )
      C*** ** NCH FERMAPS THE FILE ****
      IF( INC ) NAE * ABEY2 ) GC_TC19
      INC = INC + 7
      TEMP = SIZE*C*5 - 0.06*TNC
      TEMP = I-C*C7 + SIG*C*3C
      XT = XS + TEMP*CTH - TEMP*STH

```

```

YT = YS + TEMP*STH + TEMP*CTH
CALL SYNPCL( XT , YT , C*14 , RCD, THETA, NAC )
IF( EXP * EC < 0 ) GC = C*12
TEMP = ( TAC - 6.0 ) * C*12
XT = XT + TEMP*STH
YT = YT + TEMP*CTH
CALL SYNPCL( XT , YT , C*14 , 7H(XIC) , THETA, 7 )
XT = XT + C*48*CTH - C*37*STH
YT = YT + C*46*STH + C*67*CTH
CALL NUMBER( XT , YT , C*IC , EXP , THETA, -1 )
CONTINUE
1C
2C***** TRACE THE LINE BACKWARDS TO THE ORIGIN *****
CALL PLCT( XLA , YLA , +3 )
CALL PLCT( XLP , YLP , +2 )
DO 44 I = 1      ^_
        BUTTCW CF NEXT TIC *****
XLB = XLP - CTH
YLB = YLP - CTH
CALL PLCT( XLP , YLP , +2 )
TOP CF NEXT TIC *****
XLA = XLA - CTH
YLA = YLA - CTH
CALL PLCT( XLA , YLA , +2 )
CALL PLCT( XLE , YLE , +2 )
CONTINUE
44
RETURN
END

```

PRINTER PLCT OF AMPLITUDE FROM FOURIER COEFFICIENT TAPE

THE FIRST INPUT CARD SHOULD CONTAIN UP TO 72 CHARACTERS OF PLOT IDENTIFICATION FOR USE BY THE COMPUTER CENTER STAFF AND IS REPRODUCED ON THE PLTTER OUTPUT. THIS CARD MUST BE PRESENT WHETHER PLOTS ARE PRODUCED OR NOT. IT IS PRESENT ONLY ONCE IN THE DECK. THE FOLLOWING SET OF CARDS IS PRESENT FOR EACH FILE OF FOURIER COEFFICIENTS PROCESSED.

CONTROL CARD DECK MAKEUP

1) KFILE,KCHAN,KSTART,KSTOP,KPLOT,KRULE,KLIST,KPUNCH,LOGLIN(1015)

WHERE,

KFILE = NO. OF FILE ON FOURIER COEFFICIENT TAPE
KCHAN = NO. OF FOURIER COEFFICIENT TAPE CHANNEL DESIRED
KSTART = FIRST BLOCK TO BE INCLUDED IN ANALYSIS
KSTOP = LAST BLOCK TO BE INCLUDED IN ANALYSIS
KPLCT = 1 FOR CALCCMP PLCT
KPLCT = 0 FOR NO CALCCMP PLCT
KRULE = 1 FOR A COLUMN OF PERIODS PER INCH ON THE PLOT
KRULE = 0 IF THIS IS NOT DESIRED
KLIST = 1 FOR A PRINTER PLOT
KLIST = 0 TO SUPPRESS THE PRINTER PLOT
KPUNCH = 1 TO PUNCH A BINARY DECK OF THE INTERMEDIATE SUMS
LOGLIN = 1 TO OBTAIN A LOG VS LINEAR SPECTRUM PLOT
LOGLIN = 0 OTHERWISE
LOGLOG = 1 TO OBTAIN A LOG LOG SPECTRUM PLCT
LOGLOG = 0 OTHERWISE

2) TITLE CARD WITH UP TO 72 COLUMNS OF IDENTIFICATION FOR THE FOLLOWING CARD IS PRESENT ONLY IF LOGLIN = 1 ON CARD 1
3) ADCDE,NPDCEE,FSTART,DPINCH
(2110,2F10.01)

WHERE,

NDCDE = NO. OF FREQUENCY DECADES TO BE PLOTTED
NPDCDE = MAXIMUM DESIRED NO. OF SPECTRAL ESTIMATES PER DECADE
FSTART = ANNOTATION TO APPEAR ON FIRST FREQUENCY DECADE ON PLOT

DPINCH = NO. OF FREQUENCY DECADES TO BE PLOTTED PER INCH
THE FOLLOWING CARD IS PRESENT ONLY IF LOGLOG = 1 ON CARD 11
4) NDCDE,NPDCDE,FSTART,DPINCH,NYOCDE,CY (2110,2F10.0,110,F10.0)
WHERE,

NDCDE = NO. OF FREQUENCY DECADES TO BE PLOTTED

NPDCDE = MAXIMUM DESIRED NO. OF SPECTRAL ESTIMATES PER DECADE
FSTART = ANNOTATION TO APPEAR ON FIRST FREQUENCY DECADE ON PLOT

DPINCH = NO. OF FREQUENCY DECADES TO BE PLOTTED PER INCH

NYOCDE = NO. OF DECADES TO BE PLOTTED ON SPECTRAL DENSITY AXIS

DY = NO. OF DECADES PER INCH TO BE PLOTTED ON SPECTRAL DENSITY AXIS

INPUT TAPE IS ON LOGICAL UNIT 9

A BLANK CARD WILL TERMINATE THE RUN OR ANOTHER COMPLETE SET OF CARDS WILL DO A SECOND ANALYSIS ON THE FOURIER COEFFICIENTS DEFINED BY THE FIRST CONTROL CARD. THE FOURIER COEFFICIENT NUMBER, ITS FREQUENCY, ITS MEAN AMPLITUDE AND ITS 95 PERCENT CONFIDENCE INTERVAL ARE PRINTED. THE MEAN AND 95 CONFIDENCE INTERVAL ARE PLOTTED ON THE PRINTER. ONLY THE MEAN IS PLOTTED ON THE CALCOMP PLOTTER.
THE PROGRAM WILL HANDLE ONLY ONE CHANNEL AT A TIME FROM THE FOURIER COEFFICIENT TAPE.

ooooooooooooooooooooooo


```

15 1NSAMP1,0,1,216? TO 170
16 0,0,
17 0,0,
18 0,0,
19 0,0,
20 0,0,
21 TART(KUNIT,10USER,KBLLOCK,XSAMPL,URCHAN,KCHAN,XFREQ,JOUMY),
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

```

KMAX

KAST

FREQ XAMP KAST KERRP KRECMIX

KMAX	KAST
100	100
105	101
110	102
115	103
120	104
125	105
130	106
135	107
140	108

CF DATA POINTS IN EACH FOURIER TRANSFORM
CALLING SEQUENCE
CALL_FCINPT (NUNIT, IDUSER, KBLOCK, NSAMPL, JRCHAN, LCHN, DFREQ, JUNITS, A
1, IND)

WHERE,

NUNIT = NO. OF TAPE UNIT CONTAINING THE COCEAN TAPE
IDUSER = 9 DIGIT USER IDENTIFICATION NUMBER (RETURNED)
KBLCK = CONSECUTIVE NUMBER OF BLOCK OF COEFFICIENTS ON THE
COCEAN TAPE (RETURNED)
NSAMPL = NO. OF POINTS IN THE FOURIER TRANSFORM (RETURNED)
JRCHAN = NO. OF TRANSFORMED CHANNELS TO BE RETURNED
LCHN = FIRST LOCATION OF AN ARRAY CONTAINING THE LOCATIONS OF
THE JRCHAN CHANNELS TO BE RETURNED
DFREQ = FREQUENCY OF DIGITAL DATA TRANSFORMED (RETURNED)
JUNITS = 8 CHARACTER NAMES OF UNITS OF DATA (RETURNED)
A = ARRAY INTO WHICH COEFFICIENTS ARE TO BE RETURNED
IND = 0 FOR NORMAL RETURN
=1 FOR END OF INPUT FILE

IF THE CALLING SEQUENCE

CALL_FCINPT (-NUNIT, IDUSER, KBLCK, NSAMPL, JRCHAN, LCHN, DFREQ, A, IND)

IS EXECUTED TITLE INFORMATION ABOUT THE TAPE WILL BE PRINTED AND
ALL INFORMATION EXCEPT THE VECTOR A WILL BE RETURNED

THE ADDRESS OF THE FIRST COEFFICIENT FOR EACH OF THE JRCHAN
TRANSFORMS RETURNED WILL BE GIVEN BY (NSAMPL+2)*(KCHAN-1) +
WHERE KCHAN = NO. OF CHANNEL REQUIRED

```

SUBROUTINE FCINPT (NUNIT, IDUSER, KBLOCK, NSAMPL, JCHAN, LCHAN, DFREQ, JU
1CIMENSON MTAPE(256), ATAPE(1), A(1), LCHAN(1), JUNITS(1)
EQUIVNT = IABS(S(NUNIT))
KUNIT = IABS(S(NUNIT))
PREAD (KUNIT, END=70) MTAPE
IDUSER = PTAPE(1)
KBLOCK = PTAPE(2)
NSAMPL = PTAPE(3)
CFREC = ATAPE(6)
IND = 0
READ (KUNIT, END=70) MTAPE
X = 1
DO 30 N=1, NCDEF
NLOC = 2*N - 1
IF (K .LE. NCRSEQ1) GO TO 10
READ (K, L=1, END=70) MTAPE
K = 1
NLOC = K*KINCR + KOFF
DO 20 J=1, JINCR
JLOC = NLOC + JINCR + NOLOC
A(JLOC) = ATAPE(JLOC)
A(JLOC+1) = ATAPE(JLOC+1)
30 K = K + 1
RETURN
50 IDUSER = PTAPE(1)
KBLOCK = PTAPE(2)
NSAMPL = PTAPE(3)
IND = 0
NCDEF = PTAPE(3)/2 + 1
JCHAN = PTAPE(5)
DO 55 I=1, IJCHAN
J = JCHAN(2*I-1) = MTAPE(2*I-1)
JUNITS(2*I) = MTAPE(2*I)
55 KINCR = 2*JCHAN + 1
KOFF = - KINCR
KCRSEQ = 256/KINCR
JINCR = 2*NCDEF
WRITEE(6, 4001) IDUSER, MTAPE(3), JCHAN, DFREQ
WRITEE(6, 4002) CORITE(6, 4003) JCHAN
LSTART = 98J + 22
LSTOP = LSTART + 8

```

```

      WRITE (6,4004) (MTAPE(K),K=LSTART,LSTOP)
      WRITE (6,4005) MTAPE(2*j+119),MTAPE(2*j+120),ATAPE(j+140)

      BACKSPACE KUNIT
      RETURN
    70  IND = 1
      RETURN
    1001 FORMAT (27HOCCEAN TAPE SPECIFICATIONS/20HUSER IDENTIFICATION!!0
     1/15H TRANSFCRM SIZE 16/32H NO. OF CHANNELS/TRANSFORMED WKS,13/23H
     2SA, LING FREQUENCY WAS F10.4 IN SAMPLES/SECOND!INFORMATION/
4002 FORMAT (20HCHANNEL,13,24H FROM OCEAN TAPE CHANNEL,13)
4003 FORMAT (18HCHANNEL,13,24H FROM OCEAN TAPE CHANNEL,13)
4004 FORMAT (1H,94)
4005 FORMAT (19H UNITS OF DATA ARE *2A4/24H CALIBRATION FACTOR WAS ,1PE
116,7)
END

C THIS SUBROUTINE DRAWS A HORIZONTAL TIC ON THE PLOT AT THE POSITION
C OF THE PEN AT THE TIME OF THE CALL. THE TIC IS C.04 INCHES LONG.
C THE PEN IS RETURNED TO ITS POSITION AT THE TIME OF THE CALL.

SUBROUTINE HORTIC
CALL WHERE (X,Y)
CALL PLOT (X-0.02,Y,2)
CALL PLOT (X+C,0.02,Y,2)
CALL PLOT (X,Y,2)
RETURN
END

SUBROUTINE REPLCE (M,N)
N=N
RETURN
END

C THIS SUBROUTINE WHEN CALLED MOVES THE TAPE ON LOGICAL UNIT NUNIT
C PAST SKIP END OF FILE MARKS. THE RECORDS SKIPPED OVER MUST BE IN
C FORTRAN BINARY. IF XSKIP IS ZERO OR NEGATIVE THE ROUTINE RETURNS
C WITHOUT MOVING THE TAPE.

```

```

SUBROUTINE SKPFL (KSKIP, NUNIT)
IF (KSKIP .LT. 1) RETURN
IF (NUNIT .EQ. 1) KSKIP = 1
END = 201 I CUMMY
10 NOT DO 10 NUE
10 DO 20 NUE
20 ENDO

```

```

SUBROUTINE XSCALE (X, N, S, YMIN, DY, K)

```

```

SUBROUTINE XSCALE (X, N, S, YMAX, YMIN, TO, XY)
X = X - XY
DO 10 I = 1, N
X = X * S + TO
10 XY = XY + DY

```

```

105 INPK 5,6,6

```

```

111 MINI 7,10,10

```

```

112 2 = 0,0 1,AP,K

```

```

113 3 = 0,0 1,AP,K

```

```

114 4 = 0,0 1,AP,K

```

```

115 5 = 0,0 1,AP,K

```

```

116 6 = 0,0 1,AP,K

```

```

117 7 = 0,0 1,AP,K

```

```

118 8 = 0,0 1,AP,K

```

```

119 9 = 0,0 1,AP,K

```

```

120 10 = 0,0 1,AP,K

```

```

121 11 = 0,0 1,AP,K

```

```

122 12 = 0,0 1,AP,K

```

```

123 13 = 0,0 1,AP,K

```

20 DO UNTIL X = 1 - YMIN / 10
21 C
22 IF THE RANGE OF TIME, GOES TO MUCH
23 INCREASED TWO MUCH
24 THEN
25 X = X + 10 * YMIN / 10
26 Y = Y + 10 * YMIN / 10
27 GOTO 10
28 END
29 STOP
30 END

9

1

1. THETA, -11
1. THETA, 71

1

THE MAIN CITIES OF THE UNITED STATES

三

20***** CONTINUE LINE BACKWARDS TO THE ORIGIN *****
C***** LINE : YL8 : +2 }
C***** X : YL8 : +2 }
C***** TIC *****
C***** LINE : YL8 : +2 |
C***** X : YL8 : +2 |
C***** P : P : P : |
C***** L : L : L : |
C***** R : R : R : |
C***** A : A : A : |
C***** N : N : N : |
C***** U : U : U : |
C***** F : F : F : |
C***** O : O : O : |
C***** D : D : D : |
C***** E : E : E : |

~~UNCLASSIFIED~~

DOCUMENT CONTROL DATA - R & D

1. REPORTING ORGANIZATION Naval Postgraduate School Monterey, California		2. REPORT SECURITY CLASSIFICATION Unclassified 3. GROUP
3. TITLE (Subject): DIGITAL ANALYSIS OF TURBULENCE DATA ON THE IBM 360/67 AT THE NAVAL POSTGRADUATE SCHOOL		
4. DESCRIPTIVE NOTES (Type of report and, if applicable, date): Technical Report, July 1969		
5. AUTHOR(S), FIRM(S), OR OTHER SOURCE J. R. Wilson N. E. J. Boston W. W. Denner		
6. REPORT DATE 1 July 1969	7a. TOTAL NO. OF PAGES 79	7b. NO. OF REPS 3
8. CONTRACT OR GRANT NO NAFORD CONTRACT ORD TASK-93C-005/ PROJECT NO 551-1/UR104-03-01	9a. ORIGINATOR'S REPORT NUMBER(s) NPS-58UW9071A	
9b. OTHER REPORT NUMBER(S) (Any other numbers that may be assigned this report)		
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.		
11. SUPPLEMENTARY NOTES This project funded by: Naval Ordnance Systems Command	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California	
13. ABSTRACT A System of time series programs used by the Institute of Oceanography of the University of British Columbia was made available to the Department of Oceanography of the Naval Postgraduate School in February 1969. This report summarizes the system and outlines the procedures to be followed in using the programs.		
14. SYSTEM DESCRIPTION The system consists of three programs labelled UBC FTOR, UBC SCOR and UBC FCPILOT. The program UBC FTOR computes Fourier coefficients from selected channels of analog-to-digital tape and writes them on another tape. The program UBC SCOR reads the tape produced by UBC FTOR and from the Fourier coefficients calculates spectra, cospectra and quadrature spectra for the channels indicated. These are computed for each data block. The printed output gives for each quantity the average, standard deviation and a number representing the trend over the blocks. In the case of co- and quad-spectra phase and coherence are also printed out. The program UBC FCPILOT provides a Cal-comp plot of the spectra for qualitative analysis.		
15. TESTS AND EVALUATION These programs have been tested on the IBM 360/67 of the Naval Postgraduate School and produced for a test tape the same answers as produced by the U.B.C. machine.		
16. REFERENCES A system to develop the capability to use the SDS-9300 and the associated analog computer available at the Naval Postgraduate School to digitize data to be analyzed by the time series programs is included as Appendix I.		

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

~~UNCLASSIFIED~~

Security Classification

A-31408

~~UNCLASSIFIED~~

~~SECRET CLASSIFICATION~~

ITEM NUMBER	LINE A		LINE B		LINE C	
	SCALE	UNIT	SCALE	UNIT	SCALE	UNIT
Digital process turbulence Spectral analysis						

DD FORM 1 NOV 68 1473 (BACK)

S/N 0101-807-6321

UNCLASSIFIED

Security Classification

A-31409